Best Practices and What's New in User Notifications

Session 708

Kritarth Jain, iOS Engineer
Teja Kondapalli, iOS Engineer

© 2017 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.

Notifications overview

Hidden content

Modifying content

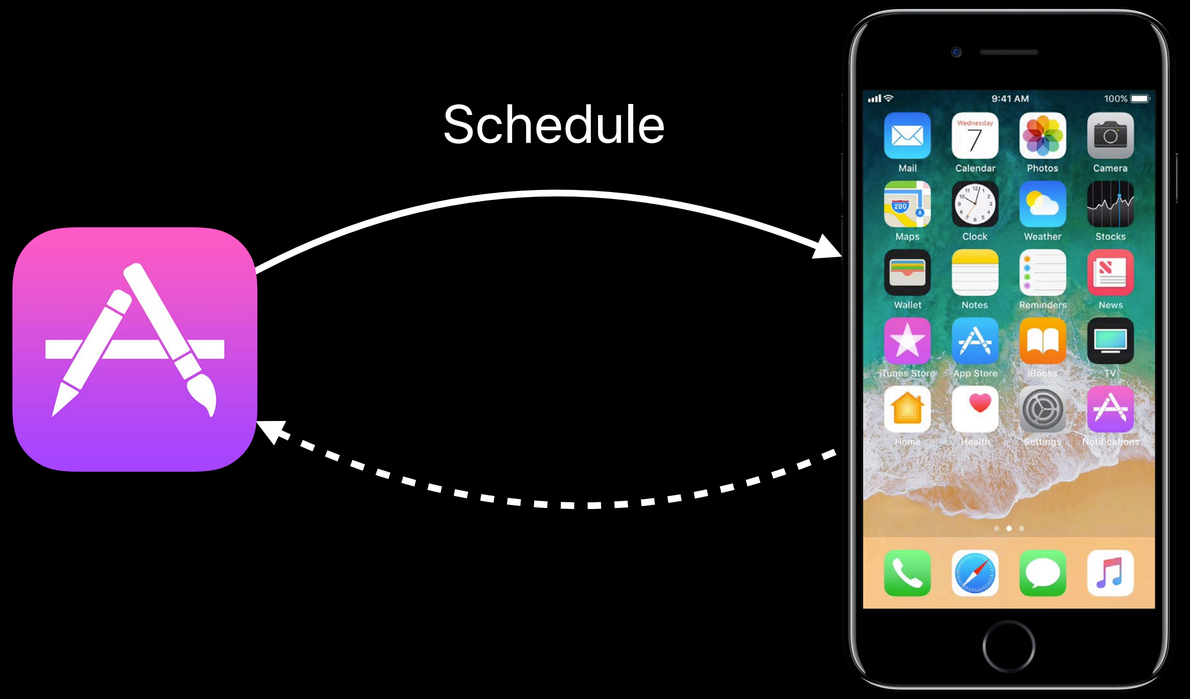
Customizing rich notifications

User input customization

Notifications Overview

Local Notifications

Local Notifications

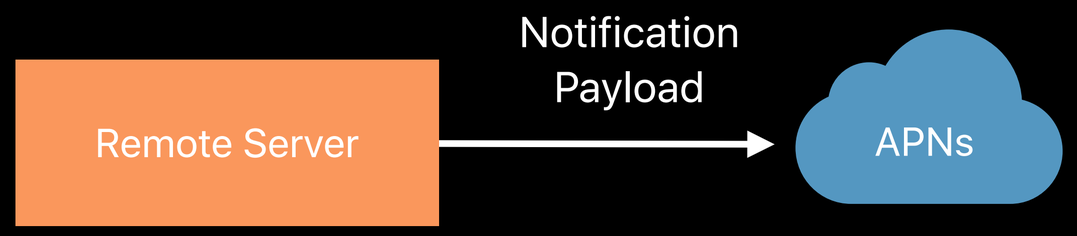


Remote Notifications

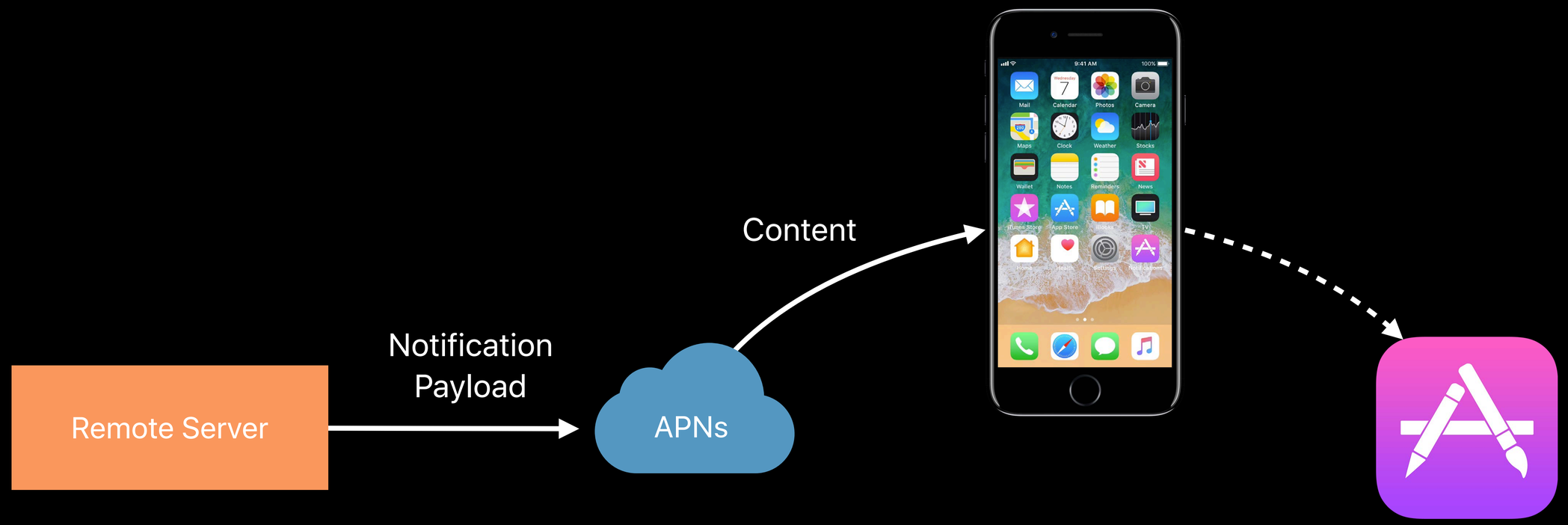
Remote Notifications

Remote Server

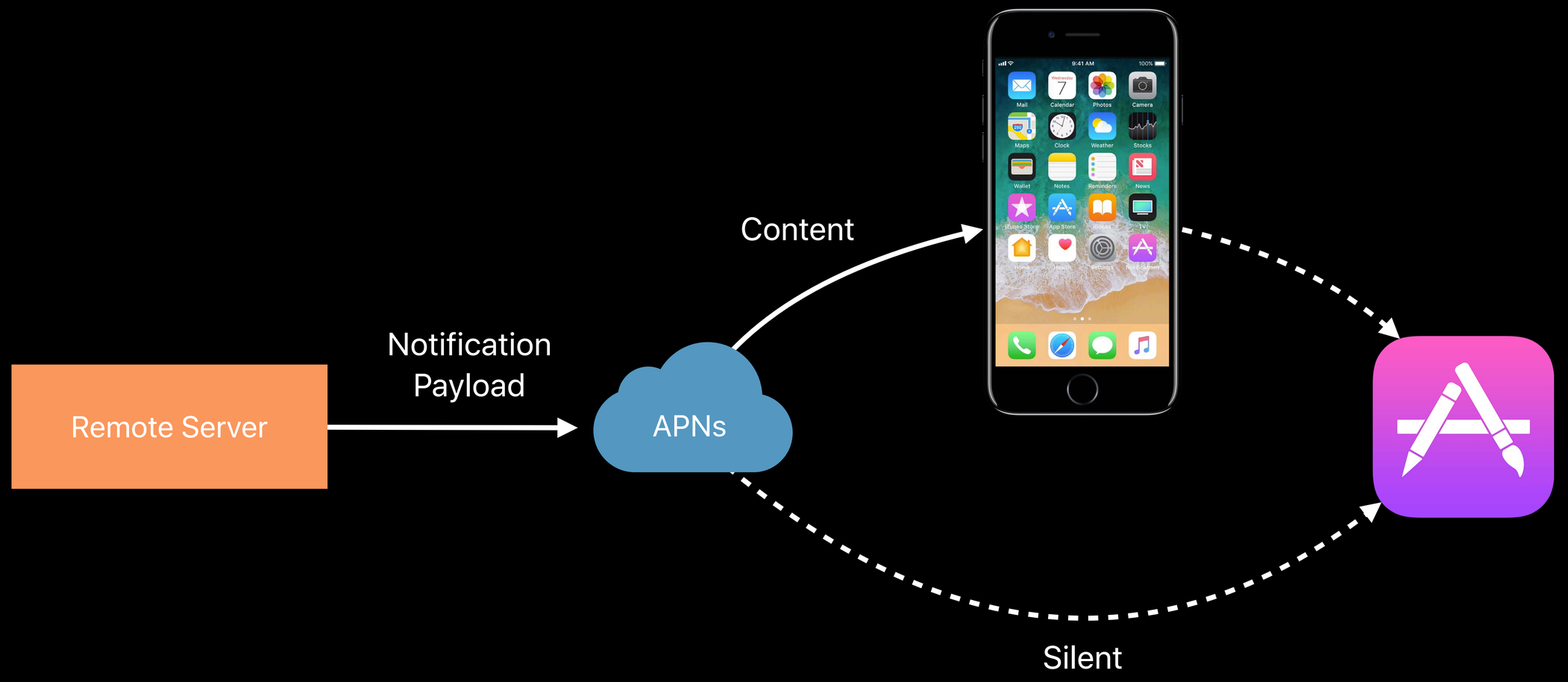
Remote Notifications



Remote Notifications



Remote Notifications

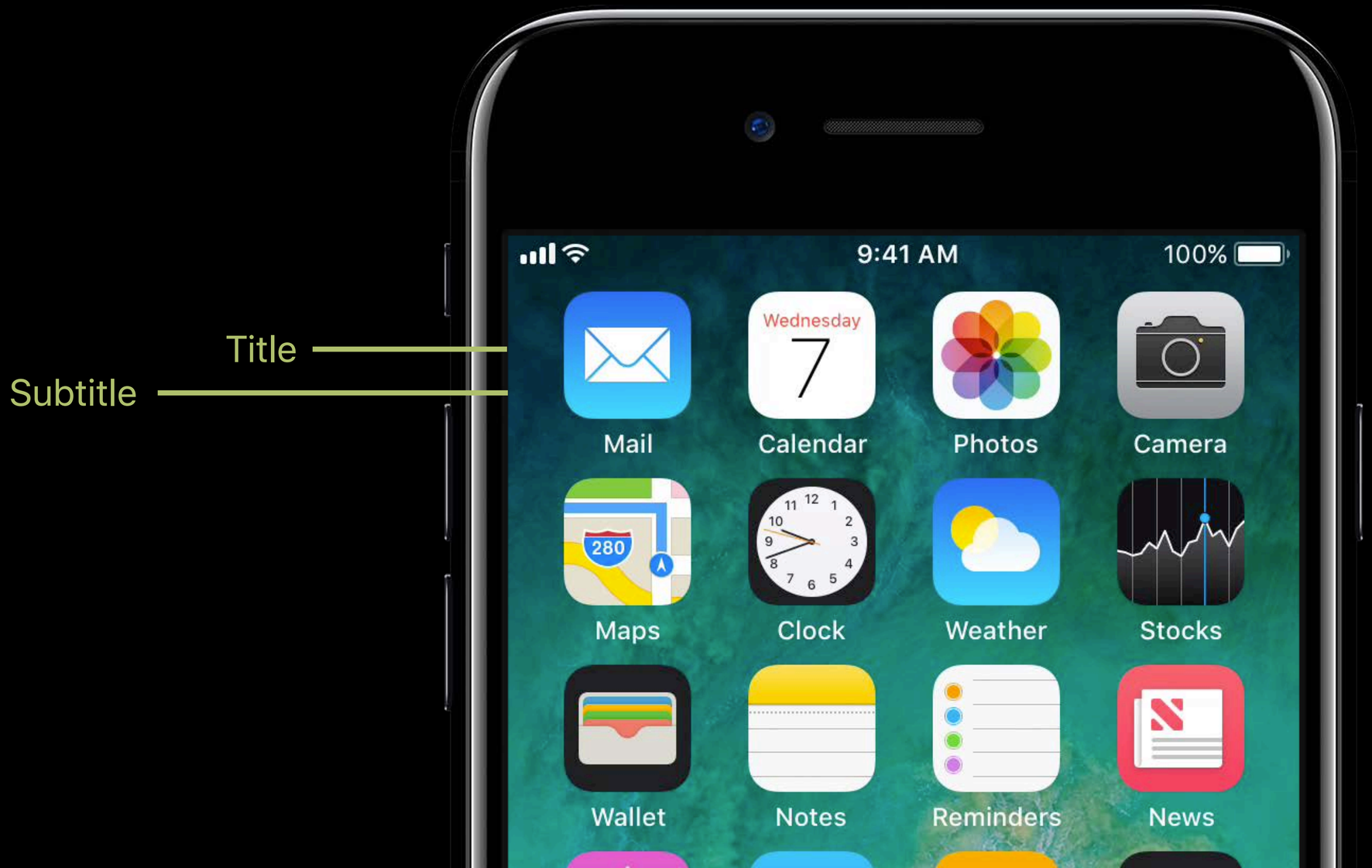


Notification Content

Notification Content

Notification Content

Notification Content



Notification Content

Notification Content

Title

Subtitle

Body

Media Attachment

Notification Triggers

Notification Triggers

Push

Notification Triggers

Push

Time Interval

Calendar

Location

Notification Presentation

Notification Presentation

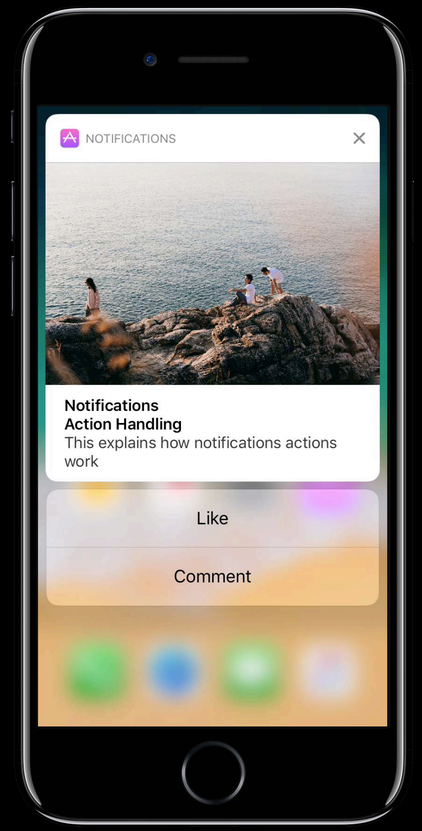
Notification Presentation

Notification Presentation

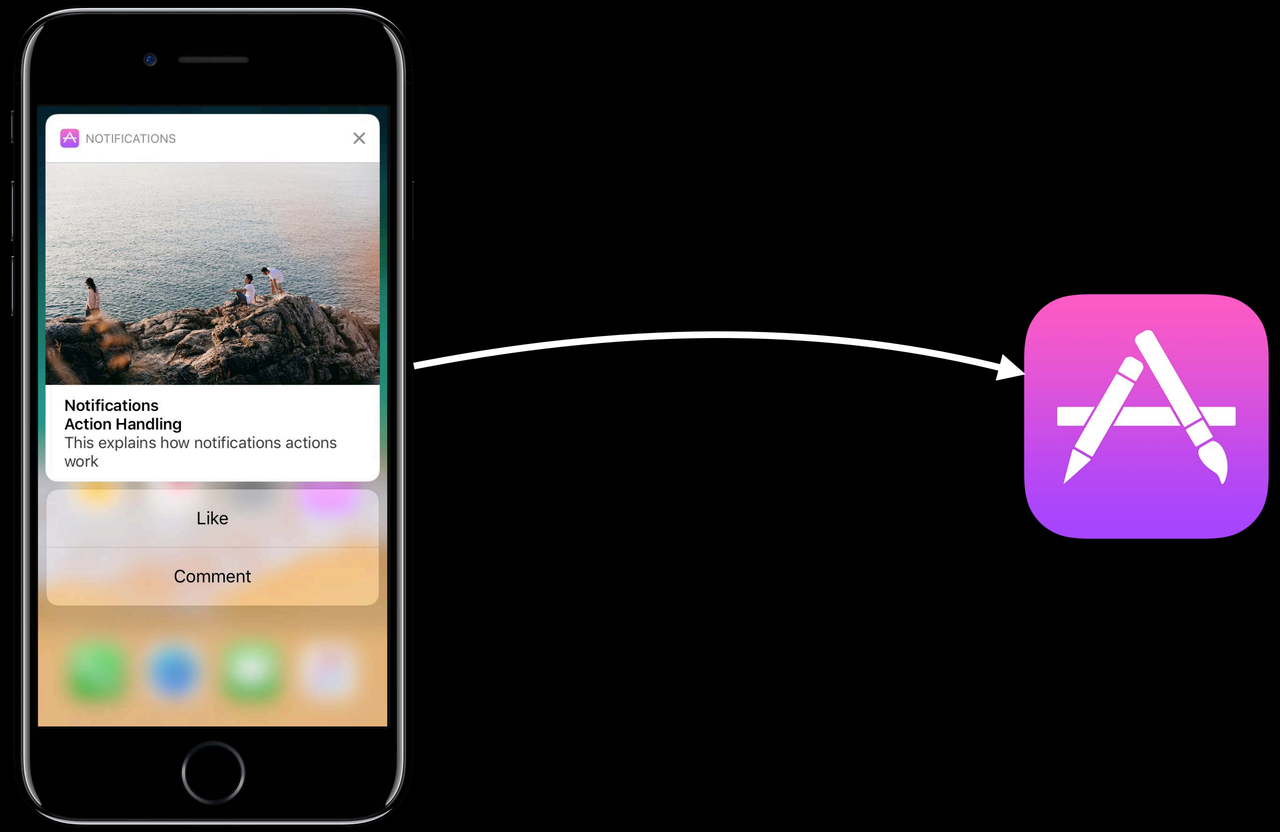
Notification Presentation

Notification Presentation

Notification Actions



Notification Actions



Notification Update

Notification Update

Notification Removal

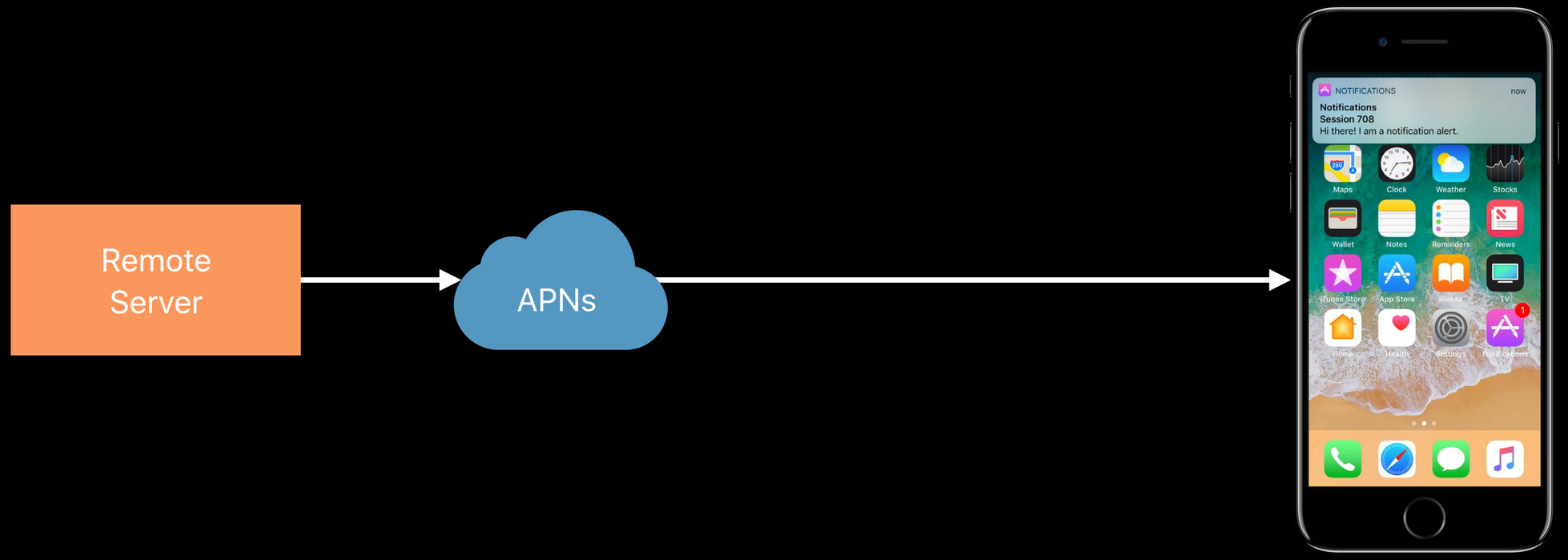
Notification Removal

Notification Extensions

Service Extension

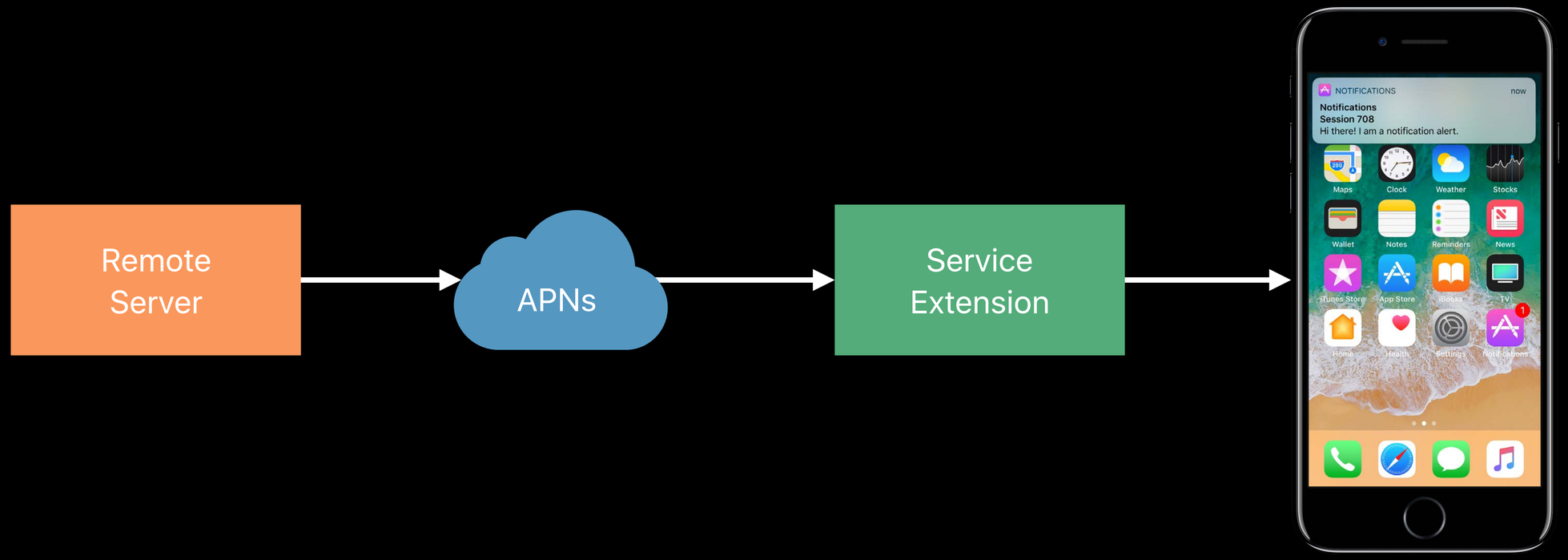
Notification Extensions

Service Extension



Notification Extensions

Service Extension

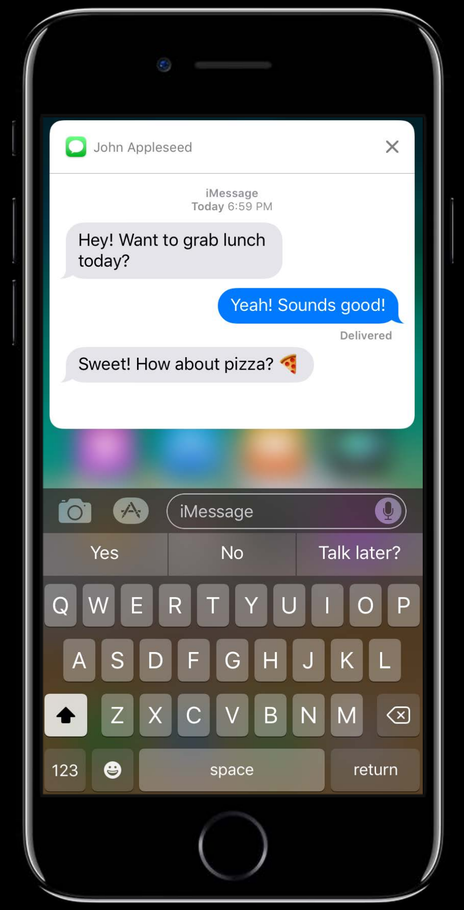


Notification Extensions

Content Extension

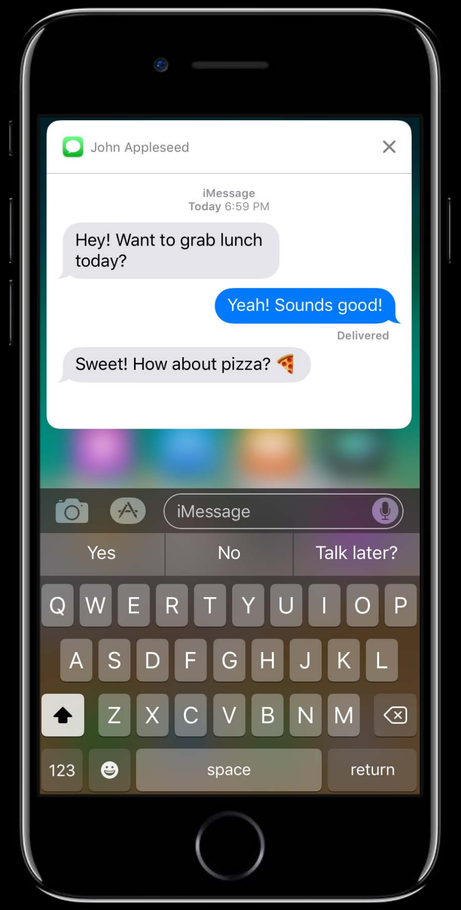
Notification Extensions

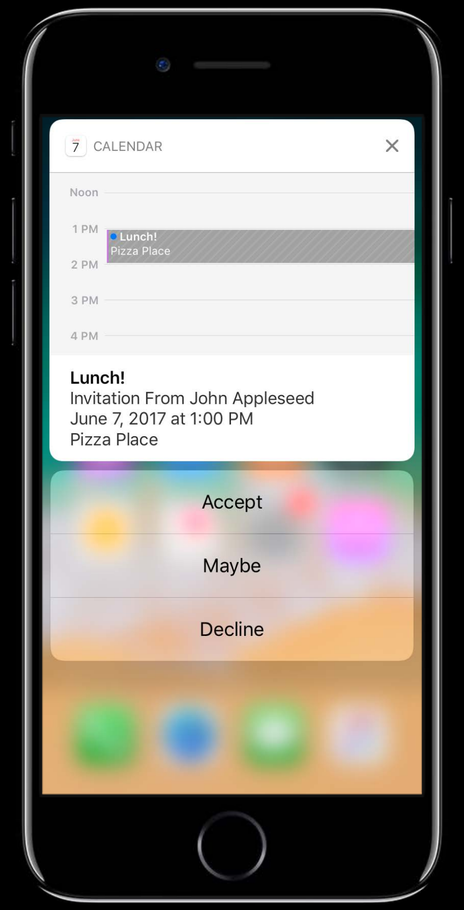
Content Extension



Notification Extensions

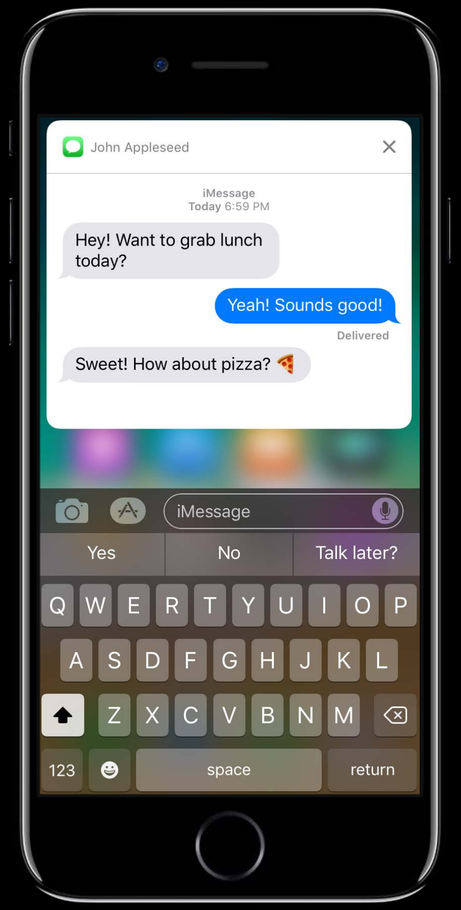
Content Extension

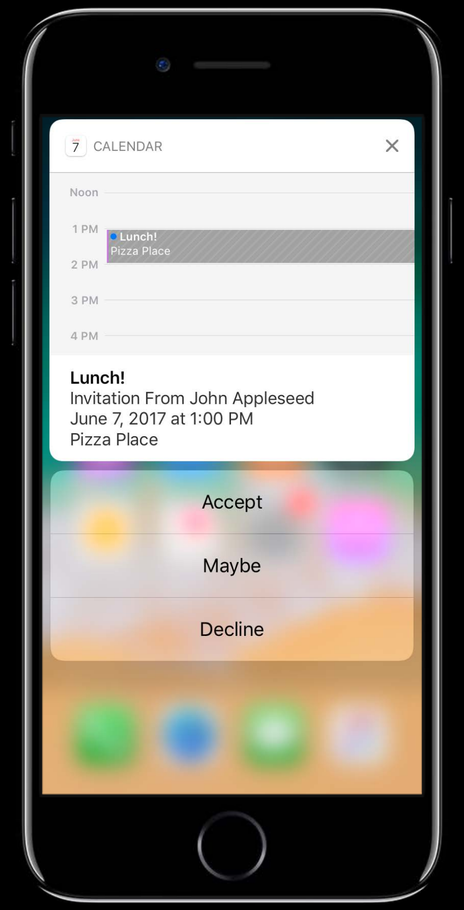


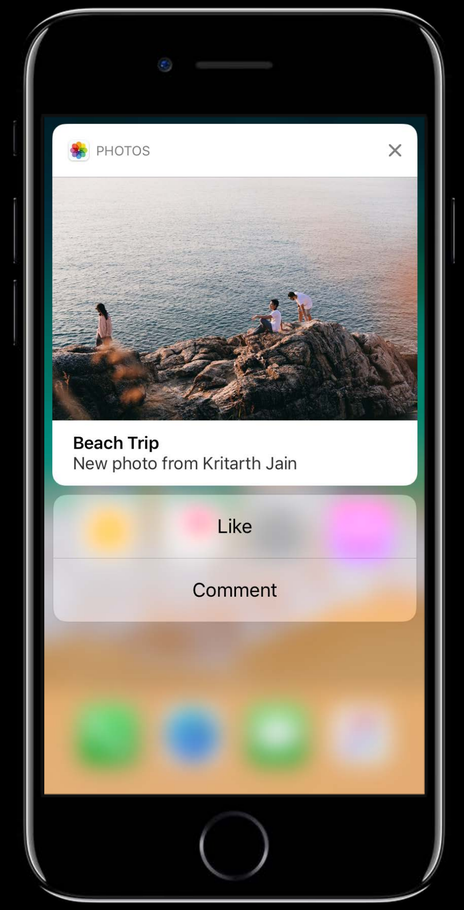


Notification Extensions

Content Extension







watchOS Notifications

watchOS Notifications

UserNotifications
UserNotificationsUI

WWDC 2016 Sessions

Introduction to User Notifications	WWDC 2016
Advanced User Notifications	WWDC 2016

Hidden Notification Content

Teja Kondapalli, iOS Engineer

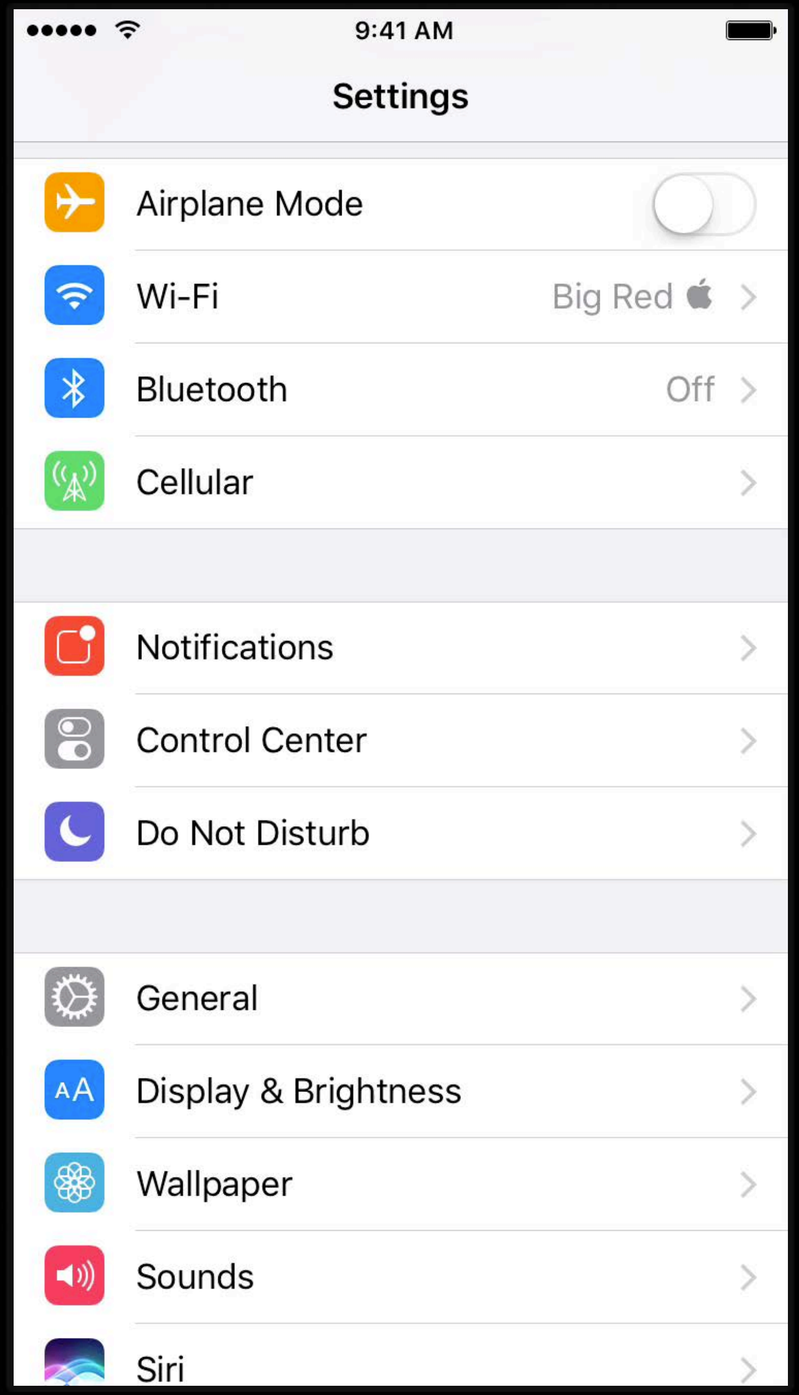
NEW

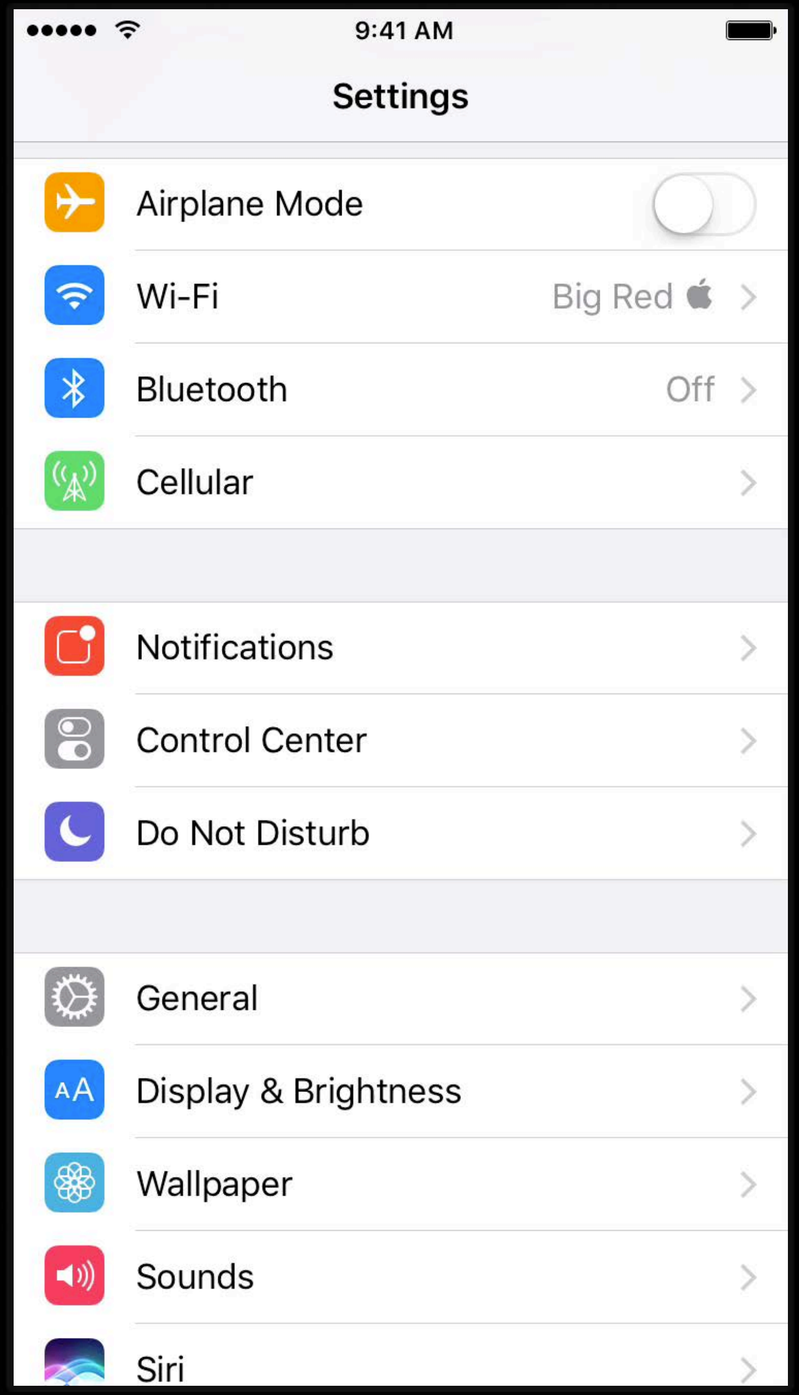
Hidden Notification Content

Teja Kondapalli, iOS Engineer

Privacy

100%
9:41
MESSAGES
now
Joey Hagedorn
Text Message
Unlock for more


9:41 AM
Settings
Airplane Mode
Wi-Fi
Big Red
Bluetooth
Off
Cellular
Notifications
Control Center
Do Not Disturb
General
Display & Brightness
Wallpaper
Sounds
Siri


9:41 AM
Settings
Airplane Mode
Wi-Fi
Big Red
Bluetooth
Off
Cellular
Notifications
Control Center
Do Not Disturb
General
Display & Brightness
Wallpaper
Sounds
Siri

100%
9:41
Wednesday, June 7
MESSAGES
now
Joey Hagedorn
Text Message
Unlock for more
Press home to unlock

Not Authenticated

100%
9:41
Wednesday, June 7
MESSAGES
now
Joey Hagedorn
Text Message
Unlock for more
Press home to unlock

100%
9:41
Wednesday, June 7
MESSAGES
now
Joey Hagedorn
Text Message
Unlock for more
Press home to unlock

Hidden Notification Content

What’s new

Hidden Notification Content

What’s new

NEW

Hidden Notification Content

What’s new

NEW

Extended support to all apps

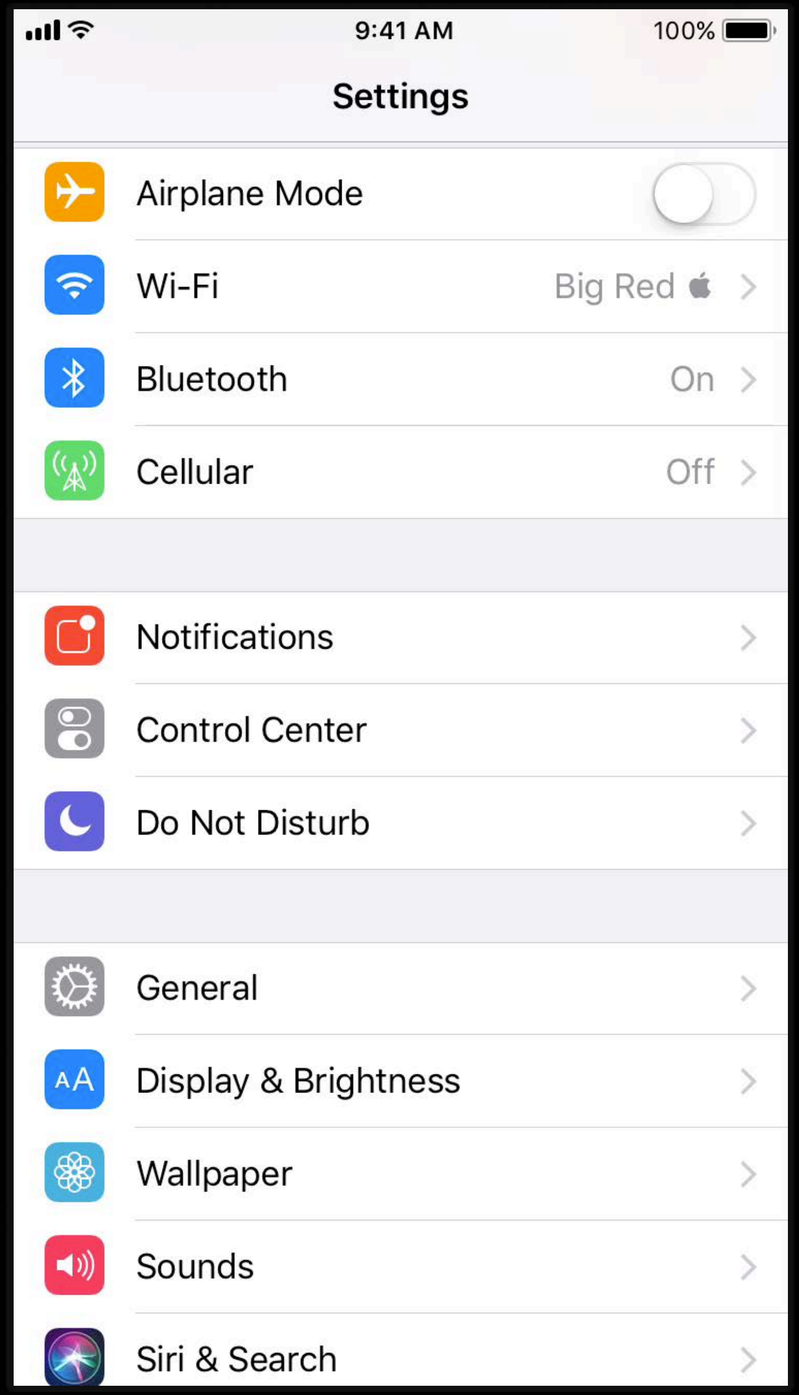
Hidden Notification Content

What’s new

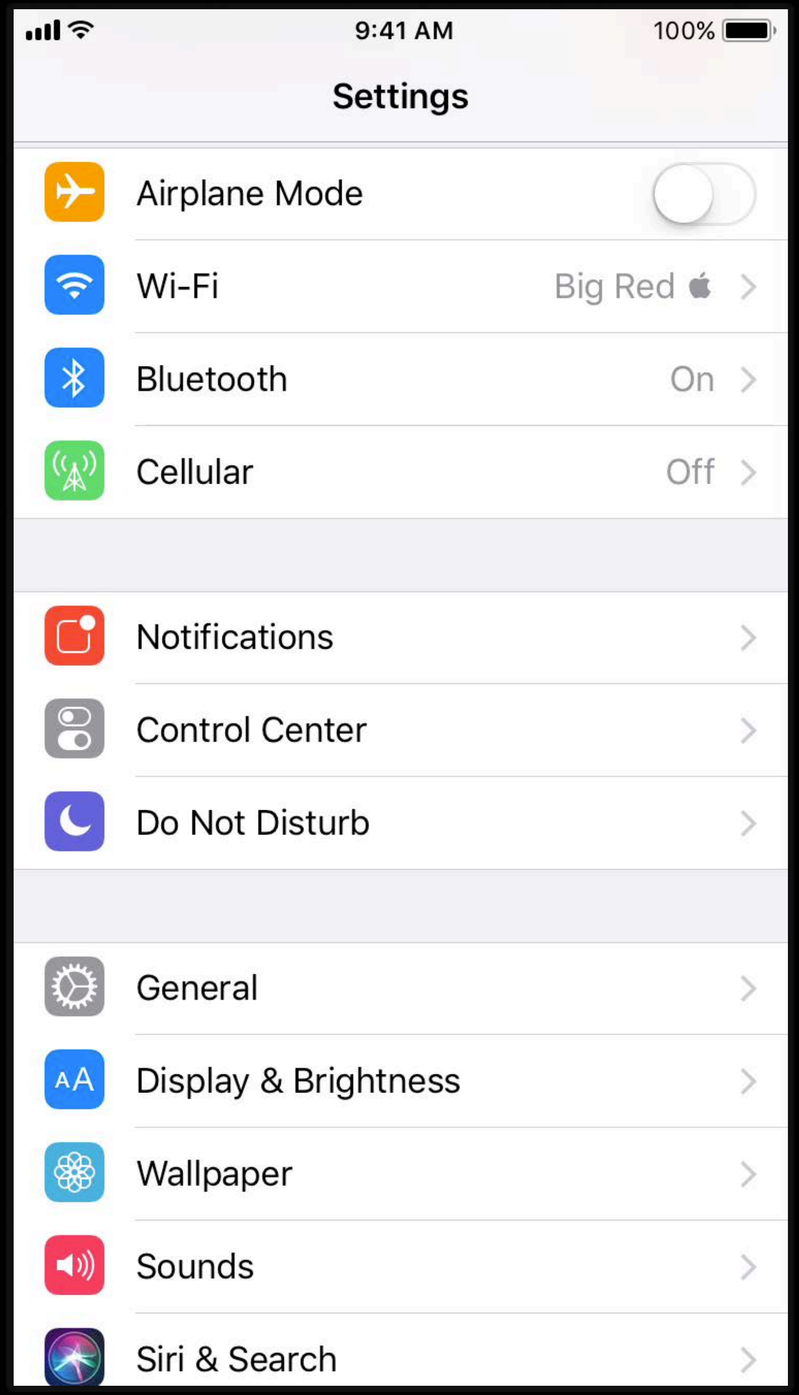
NEW

Extended support to all apps

Global setting


9:41 AM
100%
Settings
Airplane Mode
Wi-Fi
Big Red
Bluetooth
On
Cellular
Off
Notifications
Control Center
Do Not Disturb
General
Display & Brightness
Wallpaper
Sounds
Siri & Search

NEW


9:41 AM
100%
Settings
Airplane Mode
Wi-Fi
Big Red
Bluetooth
On
Cellular
Off
Notifications
Control Center
Do Not Disturb
General
Display & Brightness
Wallpaper
Sounds
Siri & Search

NEW

Hidden Notification Content

What’s new

NEW

Extended support to all apps

Global setting

Hidden Notification Content

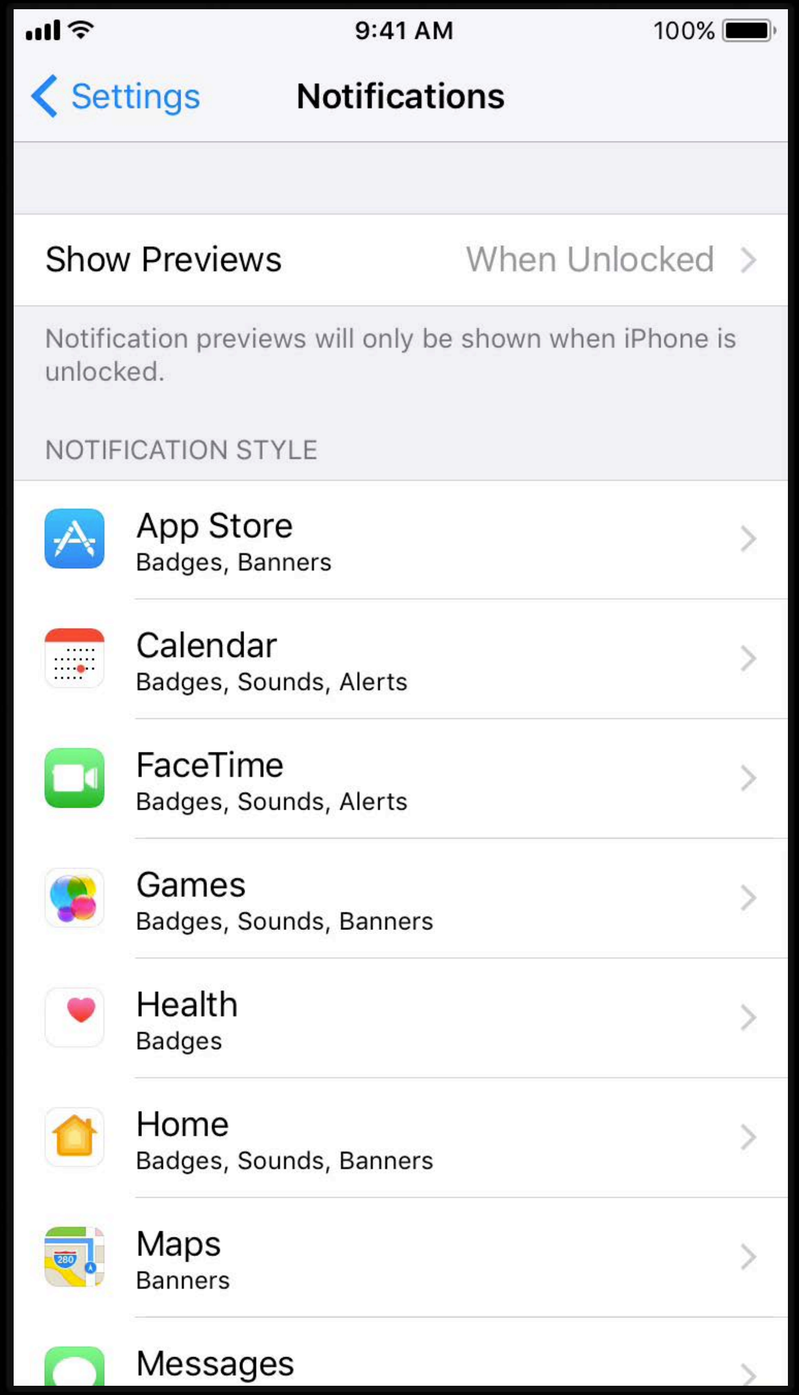
What’s new

NEW

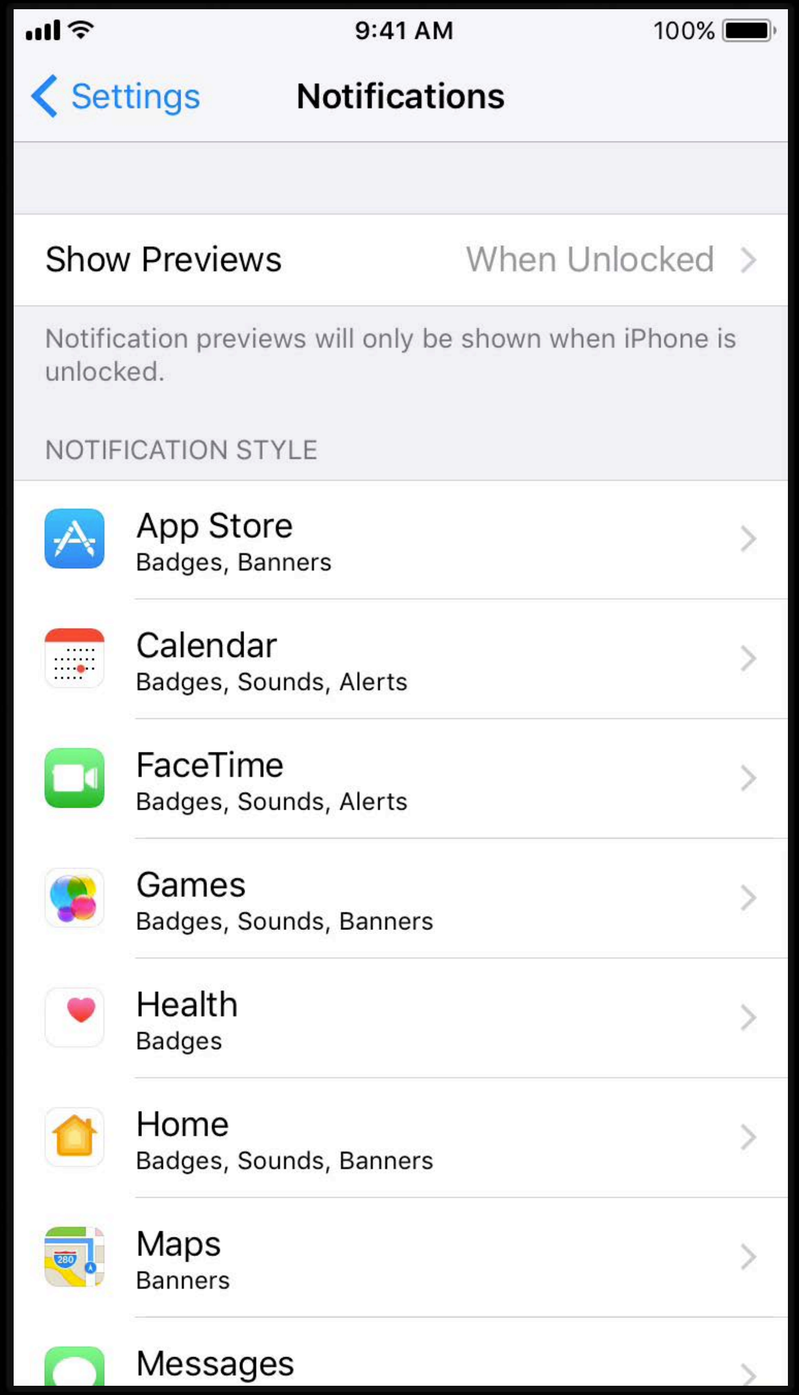
Extended support to all apps

Global setting

Settings per app


9:41 AM
100%
Settings
Notifications
Show Previews
When Unlocked
Notification previews will only be shown when iPhone is unlocked.
NOTIFICATION STYLE
App Store
Badges, Banners
Calendar
Badges, Sounds, Alerts
FaceTime
Badges, Sounds, Alerts
Games
Badges, Sounds, Banners
Health
Badges
Home
Badges, Sounds, Banners
Maps
Banners
Messages

NEW


9:41 AM
100%
Settings
Notifications
Show Previews
When Unlocked
Notification previews will only be shown when iPhone is unlocked.
NOTIFICATION STYLE
App Store
Badges, Banners
Calendar
Badges, Sounds, Alerts
FaceTime
Badges, Sounds, Alerts
Games
Badges, Sounds, Banners
Health
Badges
Home
Badges, Sounds, Banners
Maps
Banners
Messages

NEW

Hidden Notification Content

What’s new

NEW

Extended support to all apps

Global setting

Settings per app

Hidden Notification Content

What's new

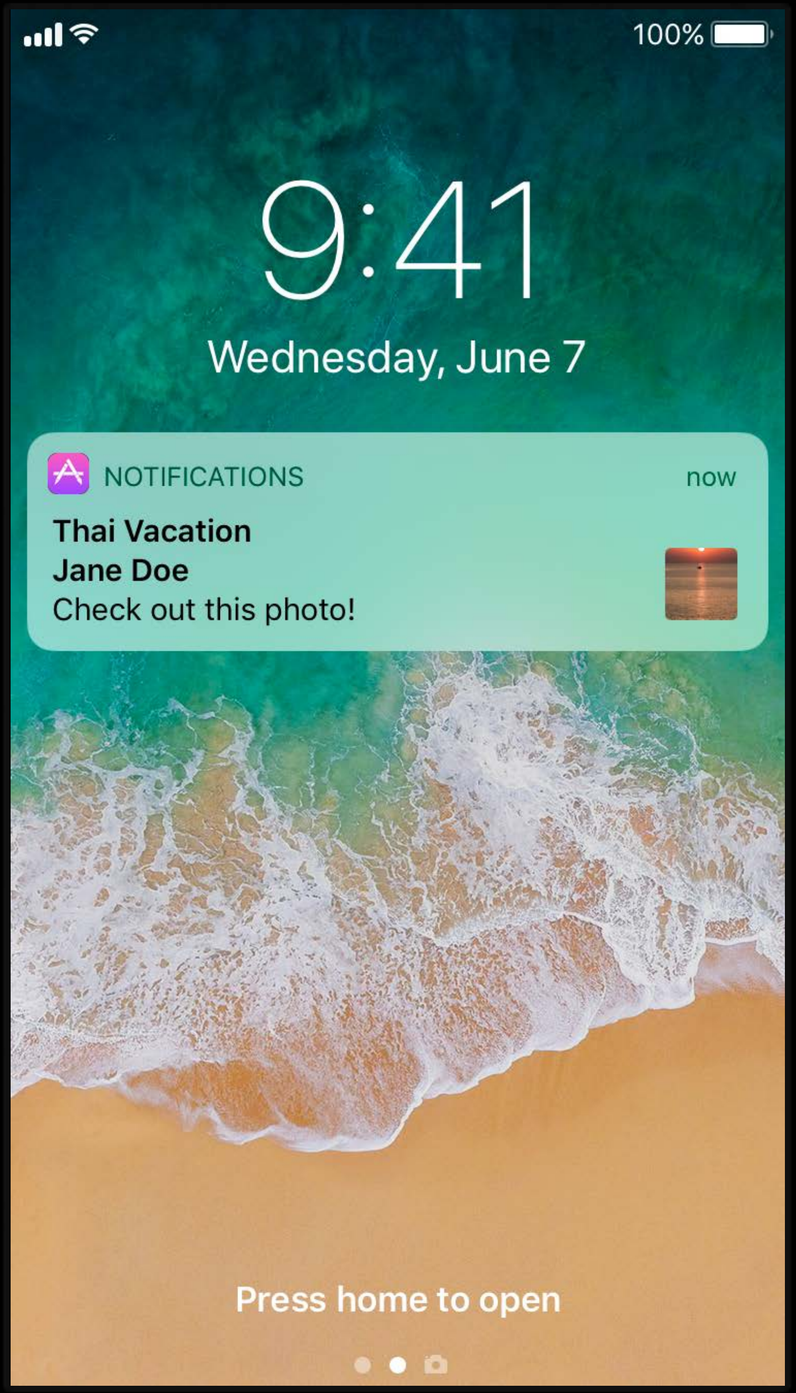
NEW

Extended support to all apps

Global setting

Settings per app

API to customize notification content

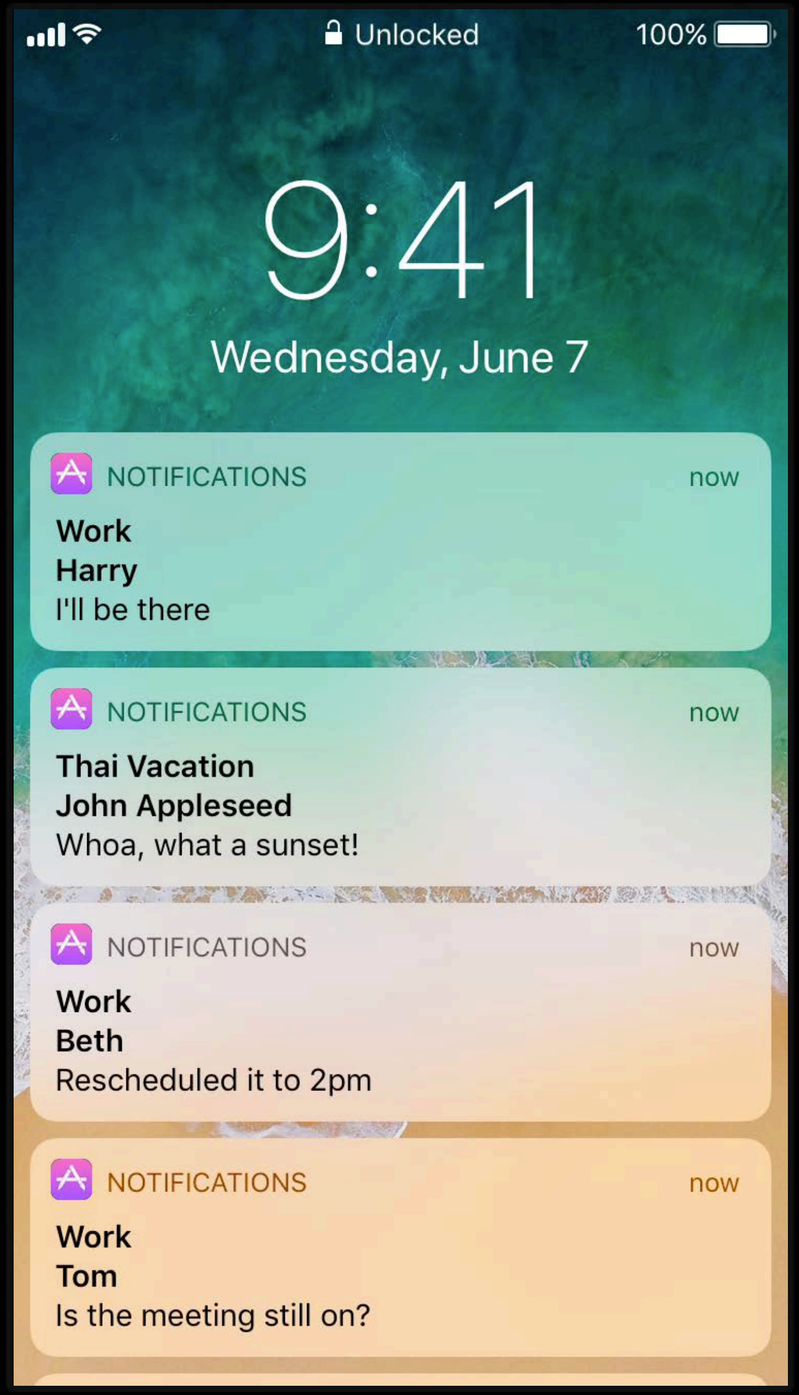

100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
Thai Vacation
Jane Doe
Check out this photo!
Press home to open

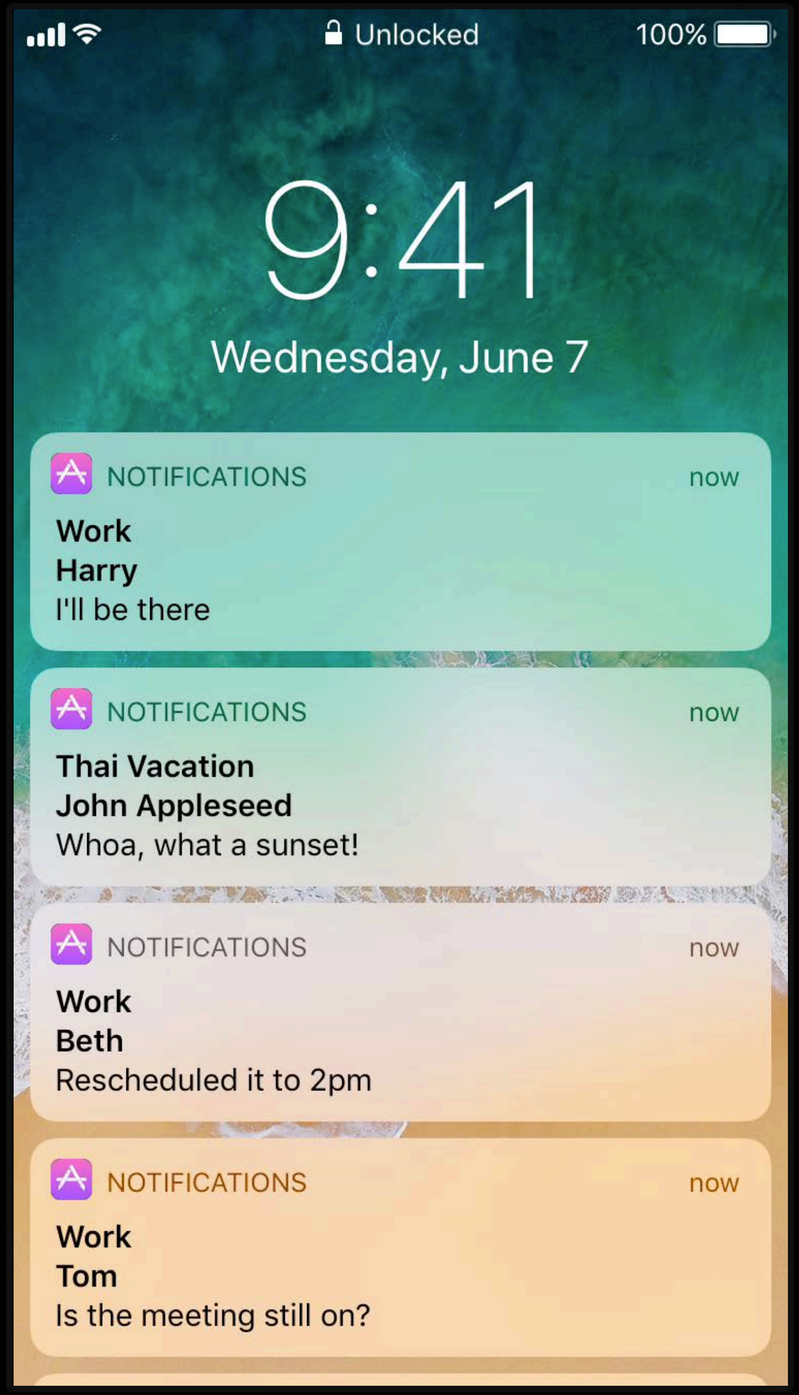
100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
Notification
Press home to unlock

NEW

Body Placeholder

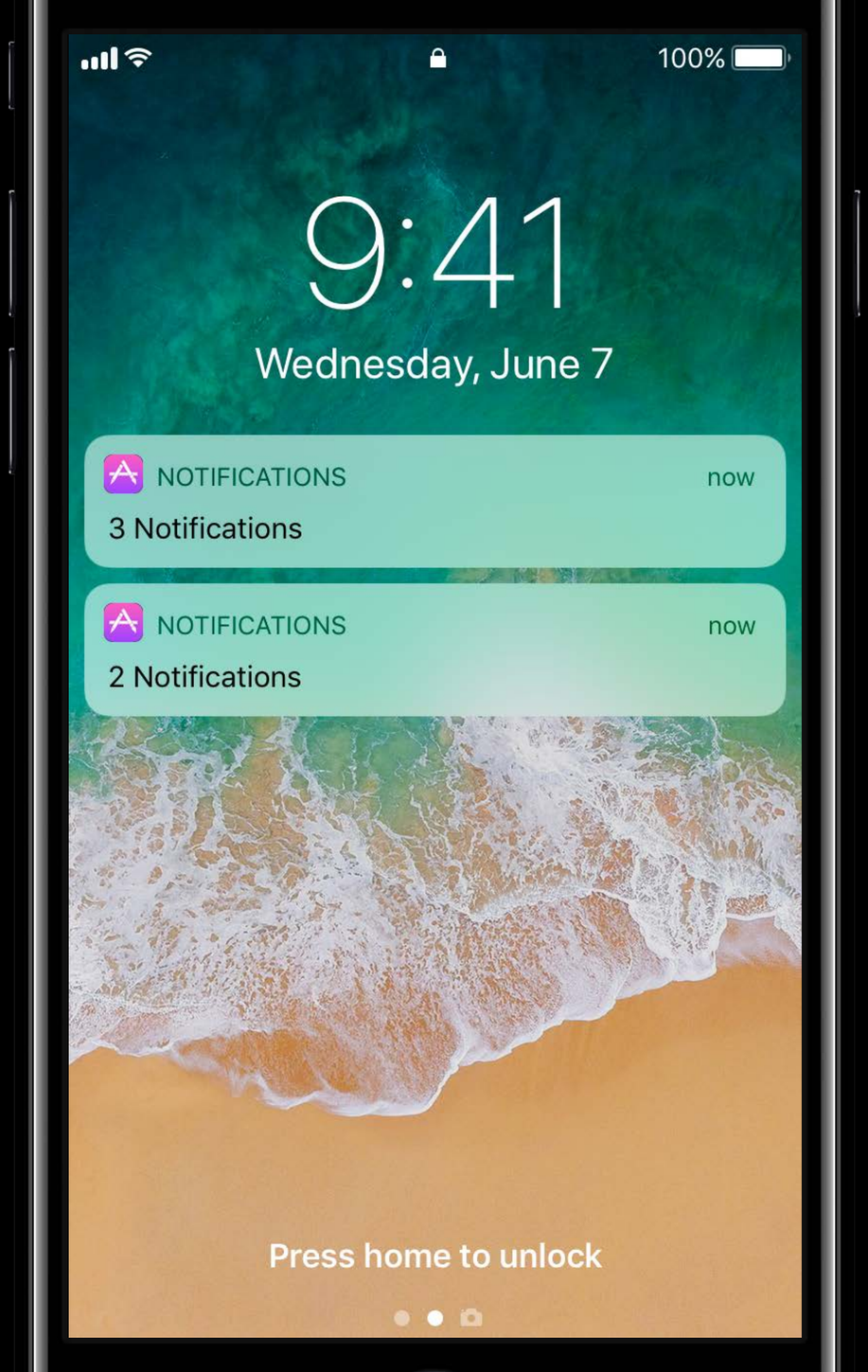
NEW


Unlocked
100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
Work
Harry
I'll be there
NOTIFICATIONS
now
Thai Vacation
John Appleseed
Whoa, what a sunset!
NOTIFICATIONS
now
Work
Beth
Rescheduled it to 2pm
NOTIFICATIONS
now
Work
Tom
Is the meeting still on?

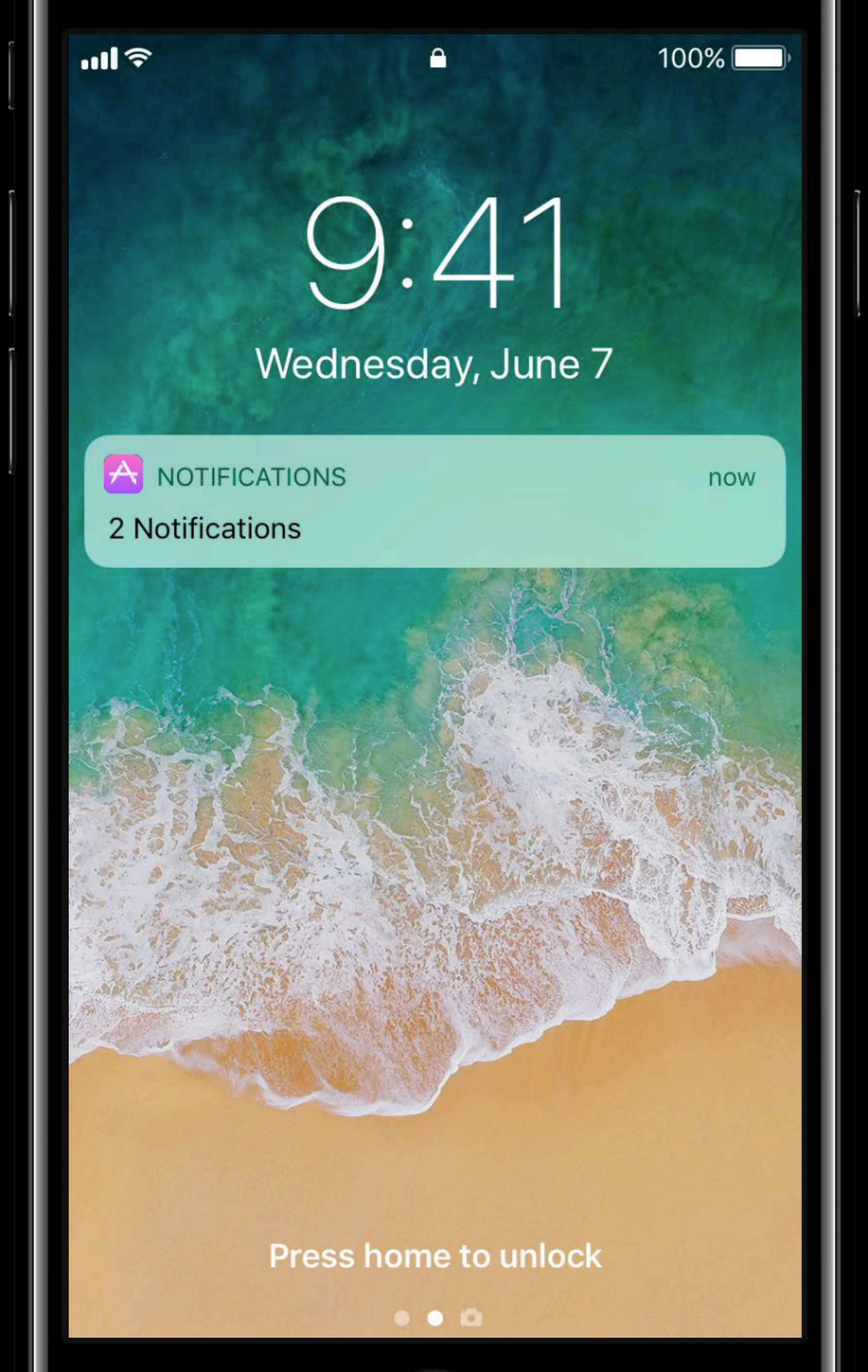

Unlocked
100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
Work
Harry
I'll be there
NOTIFICATIONS
now
Thai Vacation
John Appleseed
Whoa, what a sunset!
NOTIFICATIONS
now
Work
Beth
Rescheduled it to 2pm
NOTIFICATIONS
now
Work
Tom
Is the meeting still on?

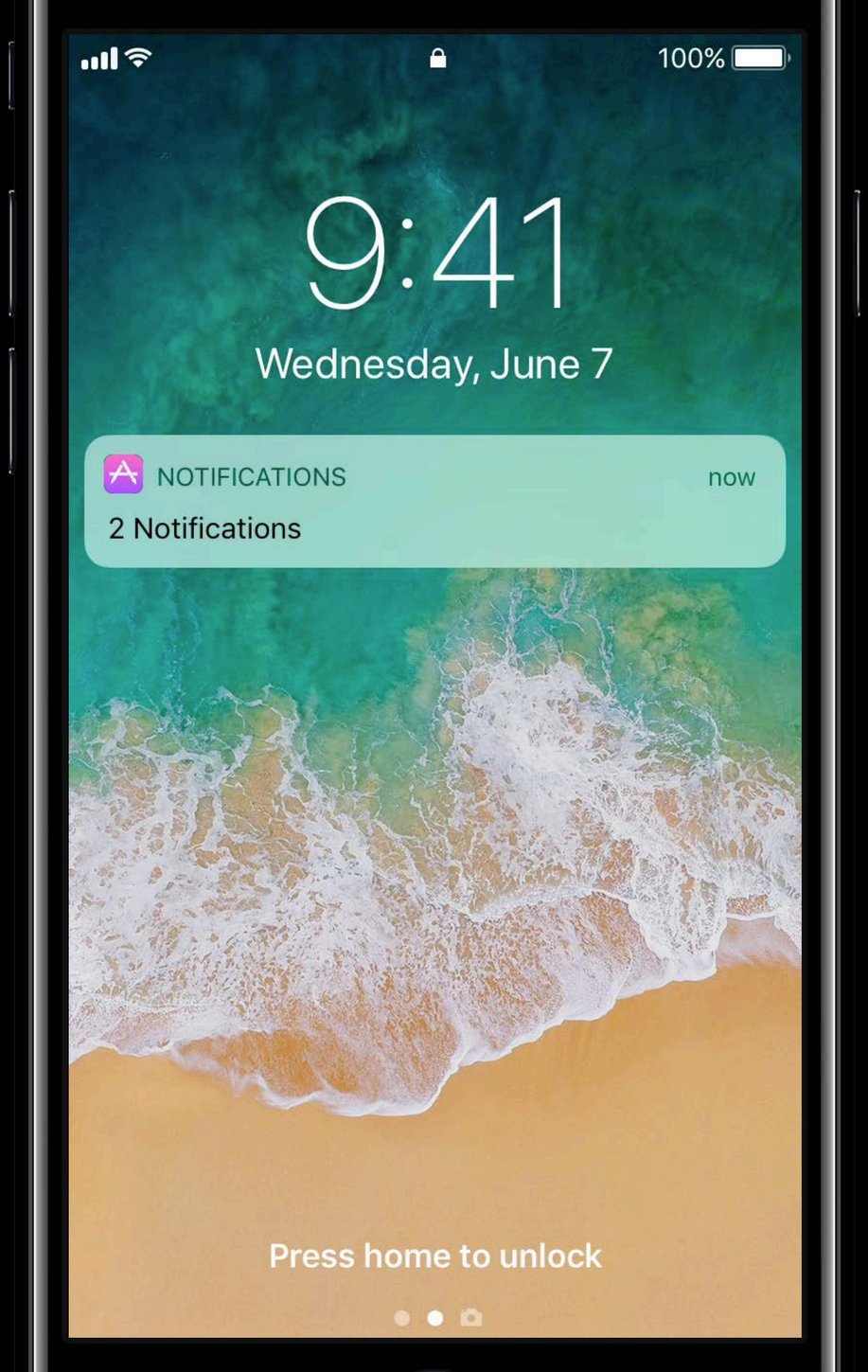
100%
9:41
Wednesday, June 7
NOTIFICATIONS now
Notification
NOTIFICATIONS now
Notification
NOTIFICATIONS now
Notification
NOTIFICATIONS now
Notification
NOTIFICATIONS now
Notification

NEW


100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
3 Notifications
NOTIFICATIONS
now
2 Notifications
Press home to unlock

NEW


100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
2 Notifications
Press home to unlock


100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
2 Notifications
Press home to unlock

NEW

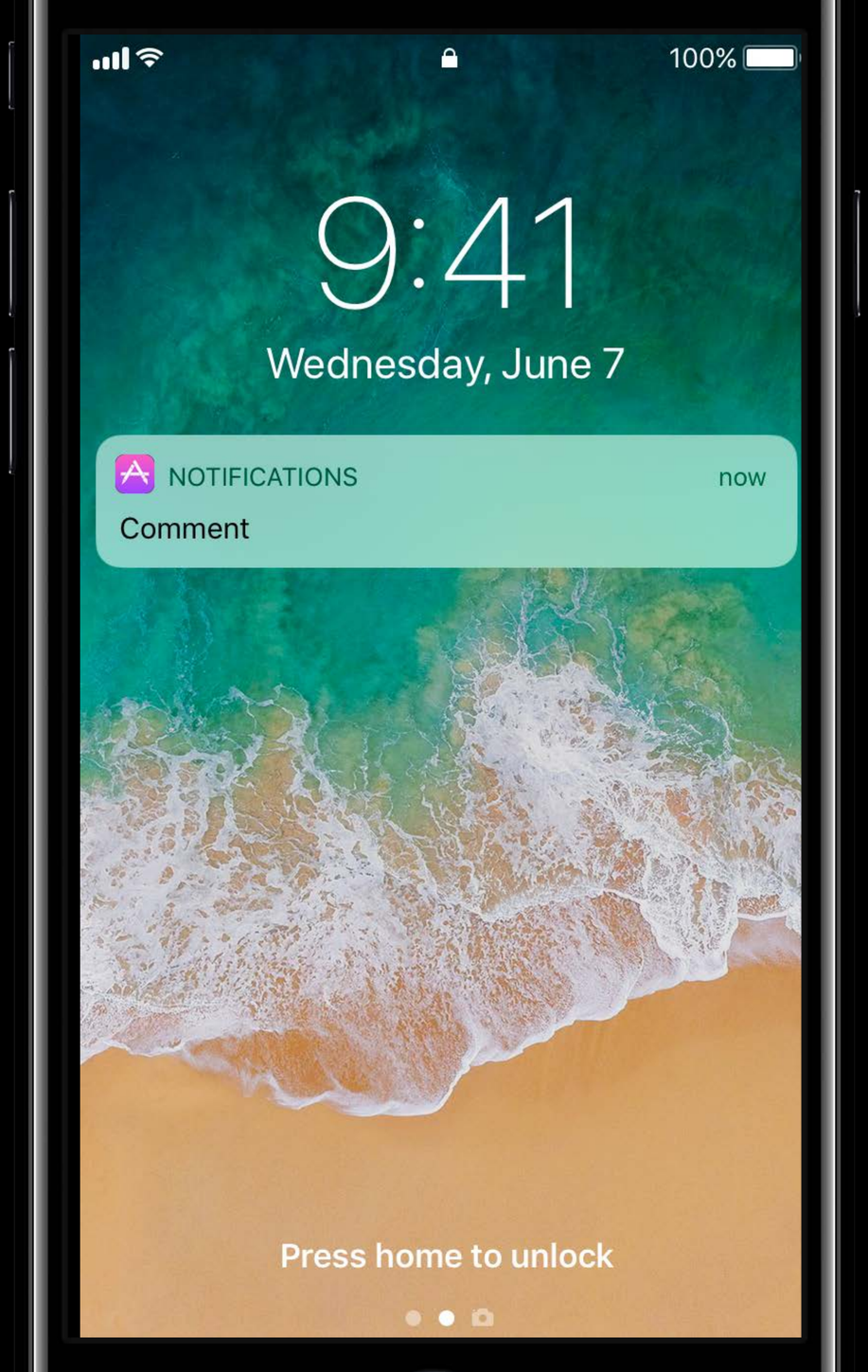
```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder: "Comment")
```

NEW

```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder: "Comment")
```

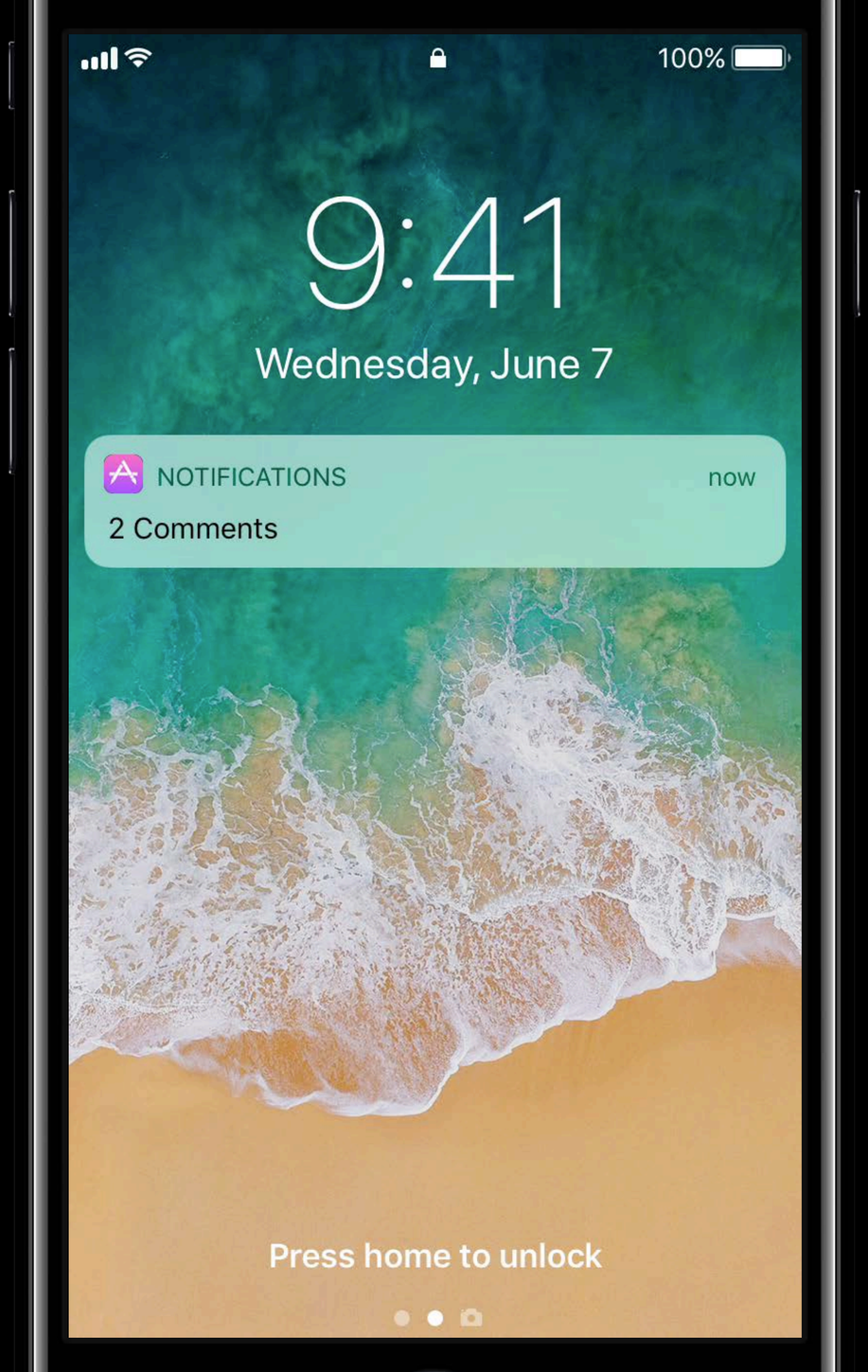
NEW

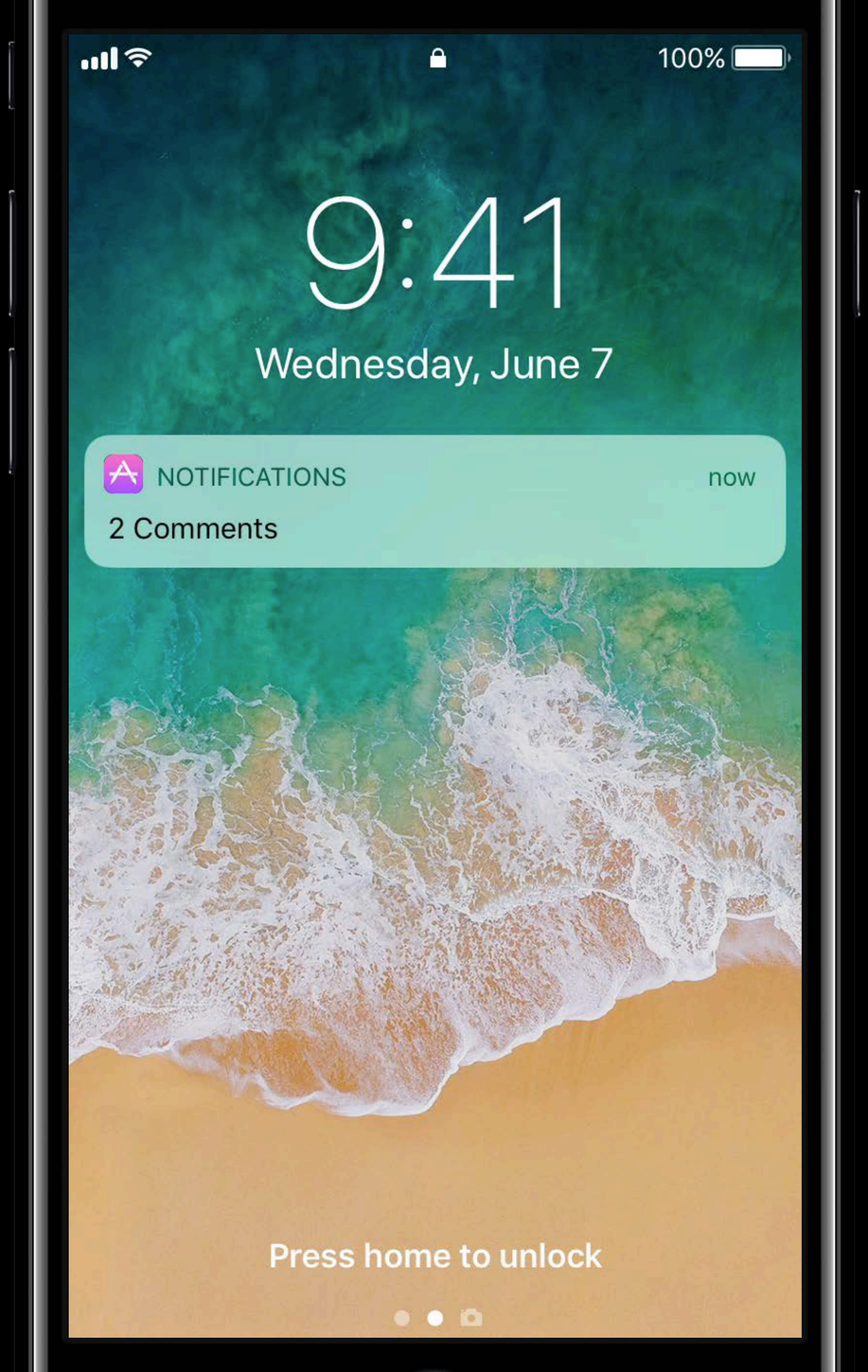
```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder: "Comment")
```


100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
Comment
Press home to unlock

NEW

```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder:"%u Comments")
```


100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
2 Comments
Press home to unlock


100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
2 Comments
Press home to unlock

Hidden Notification Content

Pluralization

ExampleNotificationApp 〉 ExampleNotificationApp 〉 Supporting Files 〉 Localizable.stringsdict 〉

Key	Type	Value
▼ Strings Dictionary	Dictionary	(1 item)
▼ COMMENT_KEY	Dictionary	(2 items)
NSStringLocalizedFormatKey	String	%#@comment@
▼ comment	Dictionary	(4 items)
NSStringFormatSpecTypeKey	String	NSStringPluralRuleType
NSStringFormatValueTypeKey	String	u
one	String	%u Comment
other	String	%u Comments

Hidden Notification Content

Pluralization

ExampleNotificationApp 〉 ExampleNotificationApp 〉 Supporting Files 〉 Localizable.stringsdict 〉

Key	Type	Value
▼ Strings Dictionary	Dictionary	(1 item)
▼ COMMENT_KEY	Dictionary	(2 items)
NSStringLocalizedFormatKey	String	%#@comment@
▼ comment	Dictionary	(4 items)
NSStringFormatSpecTypeKey	String	NSStringPluralRuleType
NSStringFormatValueTypeKey	String	u
one	String	%u Comment
other	String	%u Comments

Hidden Notification Content

Pluralization

ExampleNotificationApp 〉 ExampleNotificationApp 〉 Supporting Files 〉 Localizable.stringsdict 〉

Key	Type	Value
▼ Strings Dictionary	Dictionary	(1 item)
▼ COMMENT_KEY	Dictionary	(2 items)
NSStringLocalizedFormatKey	String	%#@comment@
▼ comment	Dictionary	(4 items)
NSStringFormatSpecTypeKey	String	NSStringPluralRuleType
NSStringFormatValueTypeKey	String	u
one	String	%u Comment
other	String	%u Comments

Hidden Notification Content

Pluralization

Localizable.stringsdict 〉 No Selection

Key	Type	Value
▼ Strings Dictionary	Dictionary	(1 item)
▼ COMMENT_KEY	Dictionary	(2 items)
NSStringLocalizedFormatKey	String	%#@comment@
▼ comment	Dictionary	(8 items)
NSStringFormatSpecTypeKey	String	NSStringPluralRuleType
NSStringFormatValueTypeKey	String	u
zero	String	لا يوجد تعليقات
few	String	u% تعليقات
one	String	تعليق واحد
many	String	u% تعليقًا
two	String	تعليقان
other	String	u% تعليقًا

Hidden Notification Content

Pluralization

Localizable.stringsdict 〉 No Selection

Key	Type	Value
▼ Strings Dictionary	Dictionary	(1 item)
▼ COMMENT_KEY	Dictionary	(2 items)
NSStringLocalizedFormatKey	String	%#@comment@
▼ comment	Dictionary	(8 items)
NSStringFormatSpecTypeKey	String	NSStringPluralRuleType
NSStringFormatValueTypeKey	String	u
zero	String	لا يوجد تعليقات
few	String	u% تعليقات
one	String	تعليق واحد
many	String	u% تعليقًا
two	String	تعليقان
other	String	u% تعليقًا

NEW

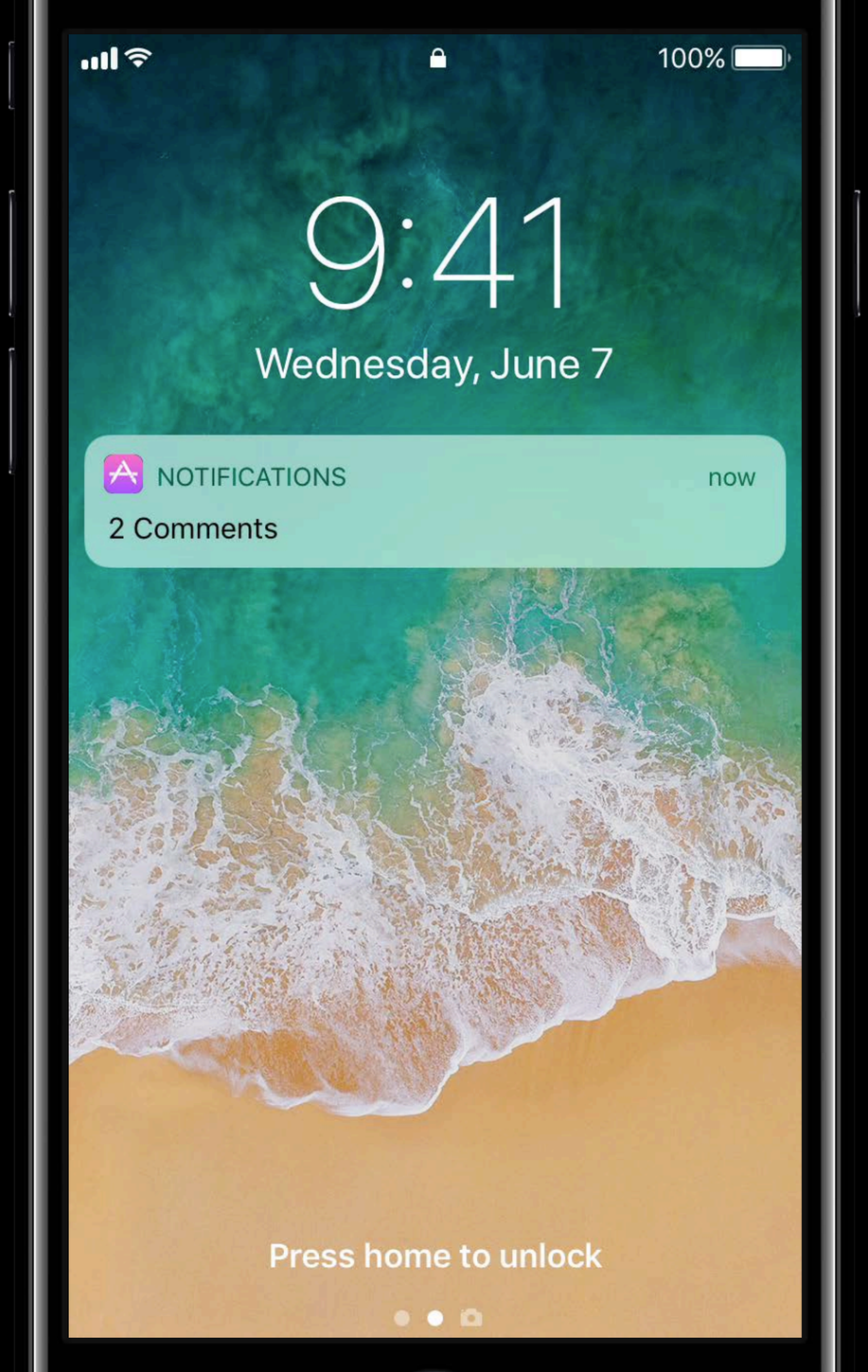
```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder:
                            NSString.localizedUserNotificationString(forKey:“COMMENT_KEY”,
arguments: nil))
```

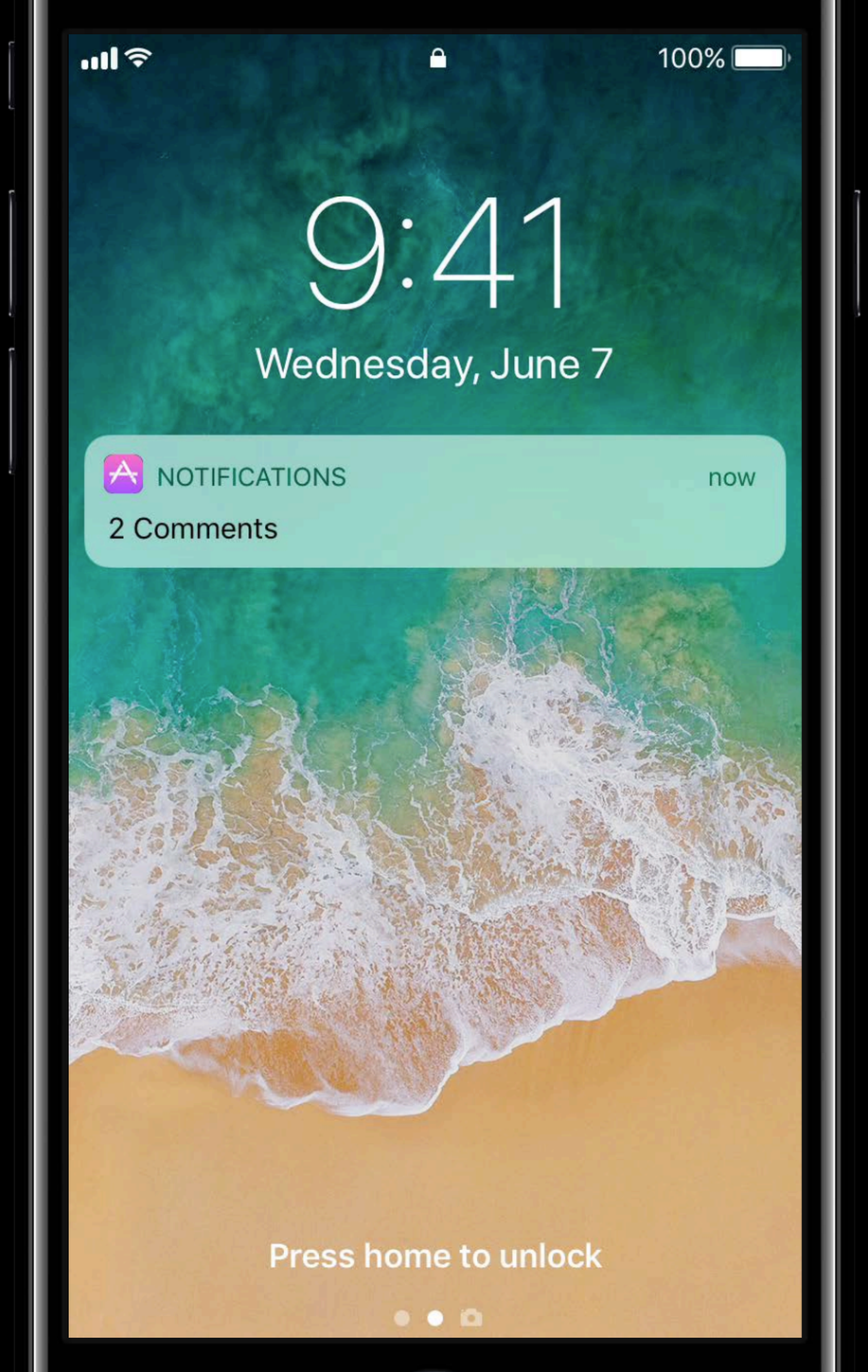
NEW

```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder:
                            NSString.localizedUserNotificationString(forKey:"COMMENT_KEY",
arguments: nil))
```

NEW

```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder:
                            NSString.localizedUserNotificationString(forKey:"COMMENT_KEY",
arguments: nil))
```

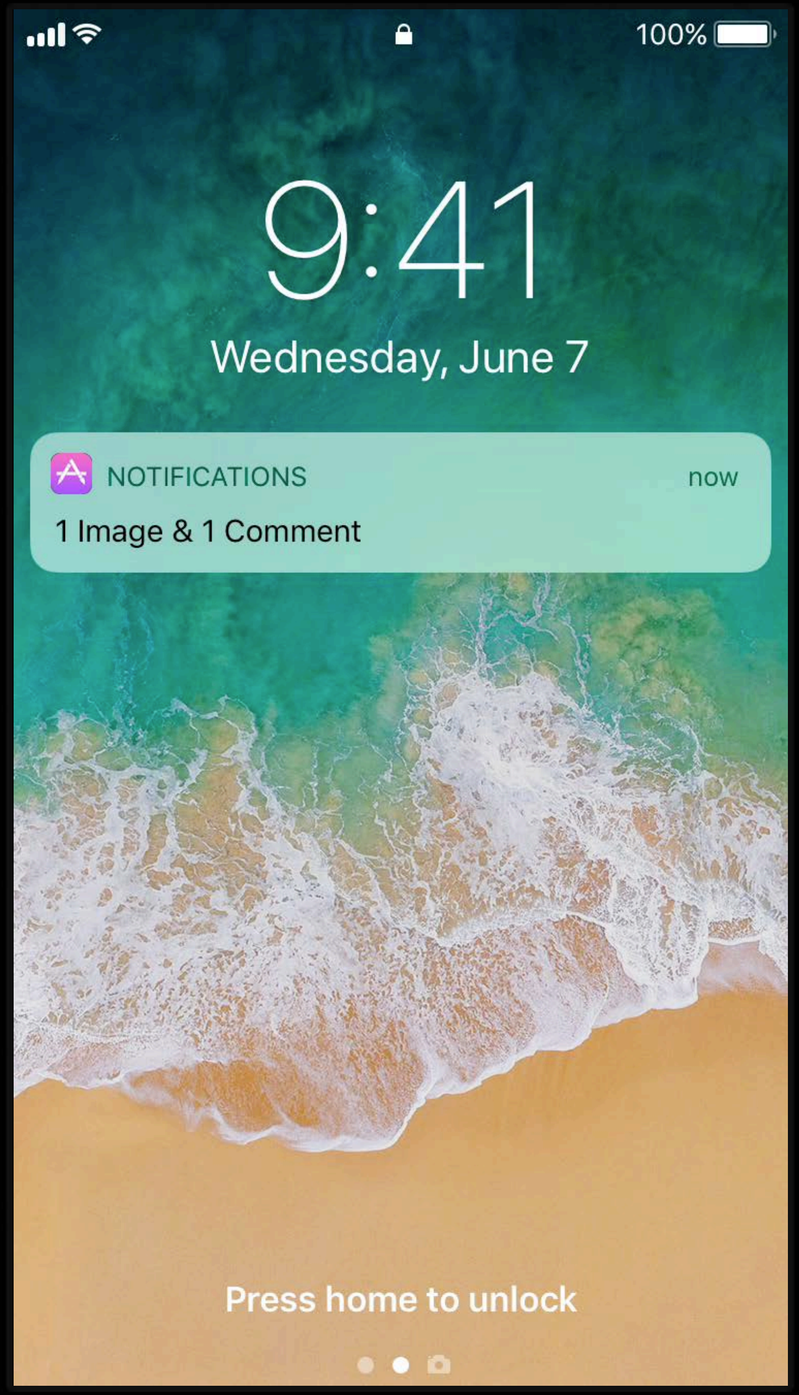

100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
2 Comments
Press home to unlock

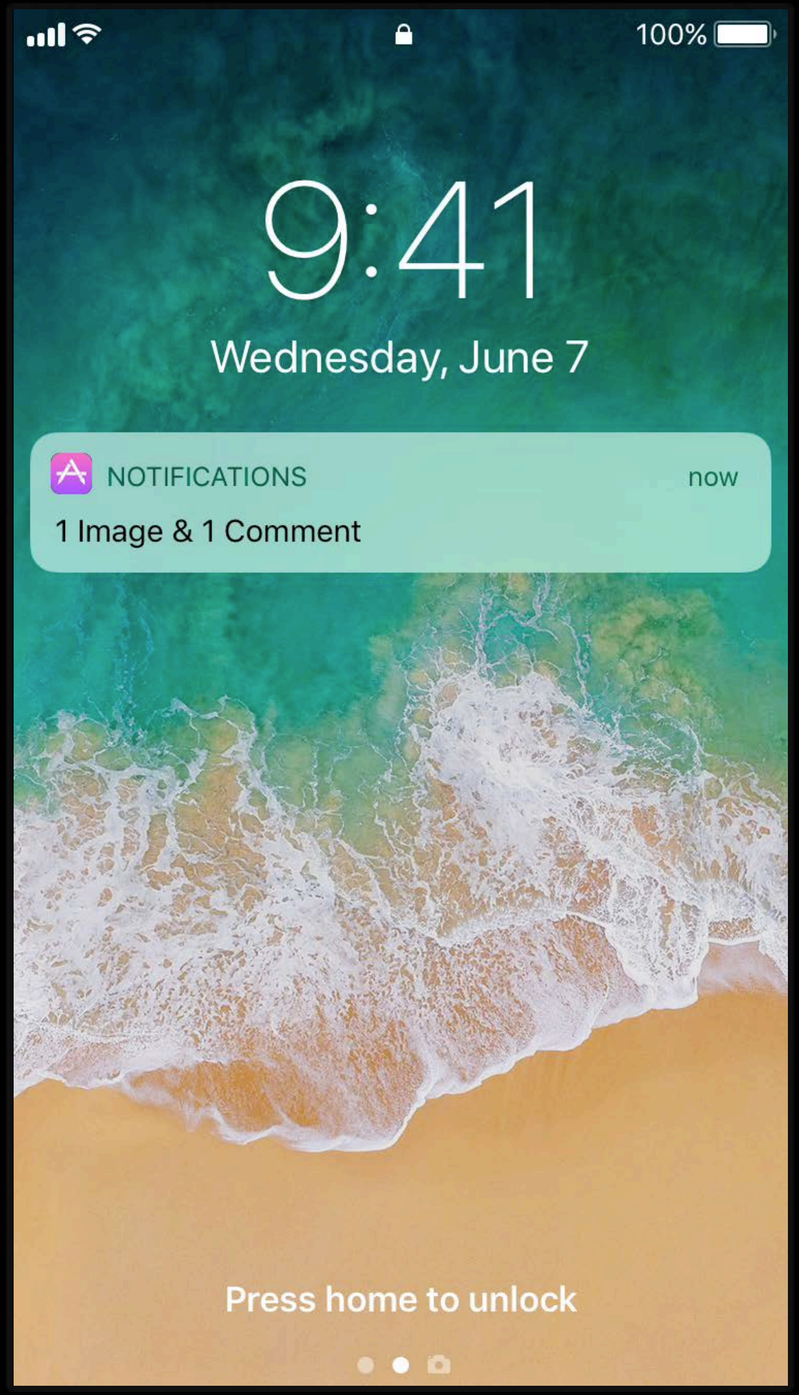

100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
2 Comments
Press home to unlock

```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                          actions: [],
                          intentIdentifiers: [],
                          hiddenPreviewsBodyPlaceholder:
                              NSString.localizedUserNotificationString(forKey:"COMMENT_KEY",
arguments: nil))

let imageCategory = UNNotificationCategory(identifier: "image-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder:
                            NSString.localizedUserNotificationString(forKey:"IMAGE_KEY",
arguments: nil))
```

```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder:
                            NSString.localizedUserNotificationString(forKey:"COMMENT_KEY",
arguments: nil))

let imageCategory = UNNotificationCategory(identifier: "image-category",
                      actions: [],
                      intentIdentifiers: [],
                      hiddenPreviewsBodyPlaceholder:
                          NSString.localizedUserNotificationString(forKey:"IMAGE_KEY",
arguments: nil))
```

```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                          actions: [],
                          intentIdentifiers: [],
                          hiddenPreviewsBodyPlaceholder:
                              NSString.localizedUserNotificationString(forKey:“COMMENT_KEY”,
arguments: nil))

let imageCategory = UNNotificationCategory(identifier: "image-category",
                        actions: [],
                        intentIdentifiers: [],
                        hiddenPreviewsBodyPlaceholder:
                            NSString.localizedUserNotificationString(forKey:“IMAGE_KEY",
arguments: nil))
```


100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
1 Image & 1 Comment
Press home to unlock


100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
1 Image & 1 Comment
Press home to unlock

NEW

```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                          actions: [],
                          intentIdentifiers: [],
                          hiddenPreviewsBodyPlaceholder:
                             NSString.localizedUserNotificationString(forKey:”COMMENT_KEY”,
arguments: nil),
                          options: [.hiddenPreviewsShowTitle])
```

NEW

```
// Hidden Notification Content

let commentCategory = UNNotificationCategory(identifier: "comment-category",
                                actions: [],
                                intentIdentifiers: [],
                                hiddenPreviewsBodyPlaceholder:
                                   NSString.localizedUserNotificationString(forKey:"COMMENT_KEY",
arguments: nil),
                                options: [.hiddenPreviewsShowTitle])
```

100%
9:41
Wednesday, June 7
NOTIFICATIONS
now
Thai Vacation
1 Image & 1 Comment
Press home to unlock

Hidden Notification Content

Best practices

Hidden Notification Content

Best practices

Hidden Notification Content

Best practices

User defined setting

Hidden Notification Content

Best practices

User defined setting

Retrieve settings using `showPreviewsSetting`

Hidden Notification Content

Best practices

User defined setting

Retrieve settings using `showPreviewsSetting`

Thread identifiers

Hidden Notification Content

Best practices

User defined setting

Retrieve settings using `showPreviewsSetting`

Thread identifiers

Pluralization

Attachments, Encryption, and More

Modifying push content

Modifying Push Content

Use cases

Modifying Push Content

Use cases

Download media attachments

Modifying Push Content

Use cases

Download media attachments

Content encryption

Modifying Push Content

Use cases

Download media attachments

Content encryption

Context specific content—location, time, health data

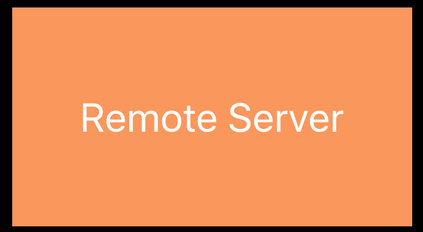
Service Extension

Modifying Push Content

Example

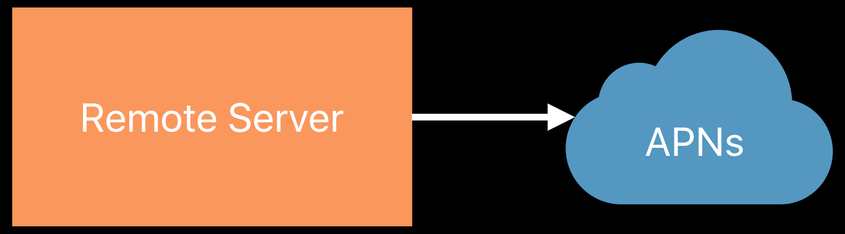
Modifying Push Content

Example



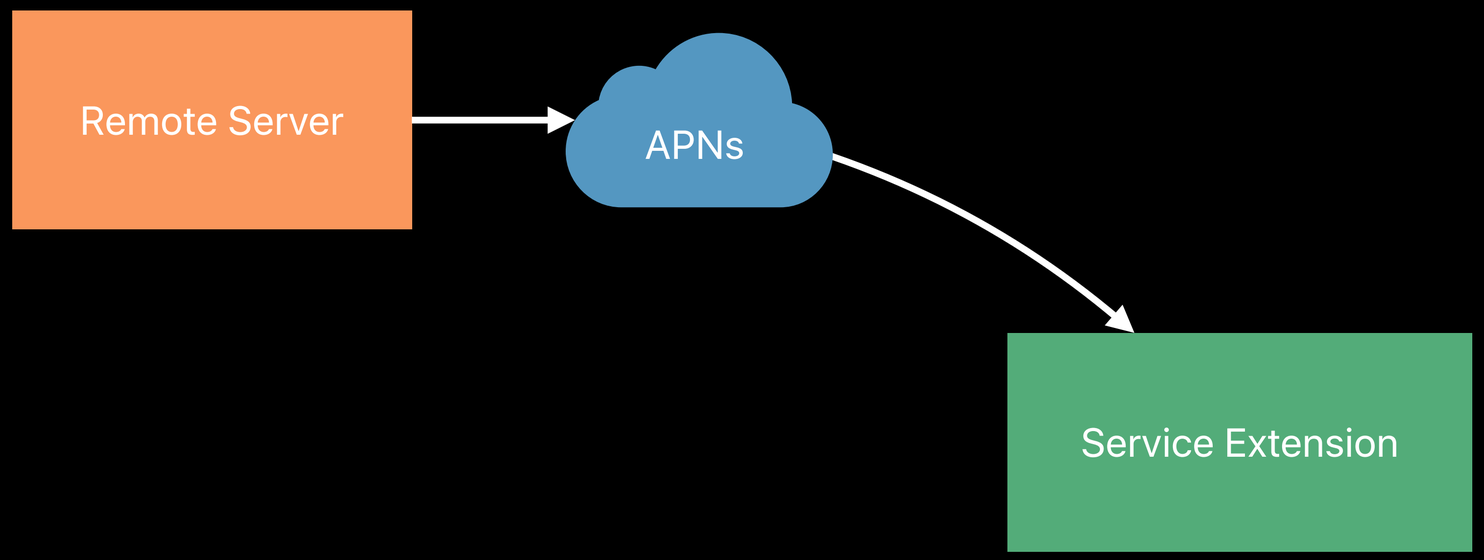
Modifying Push Content

Example



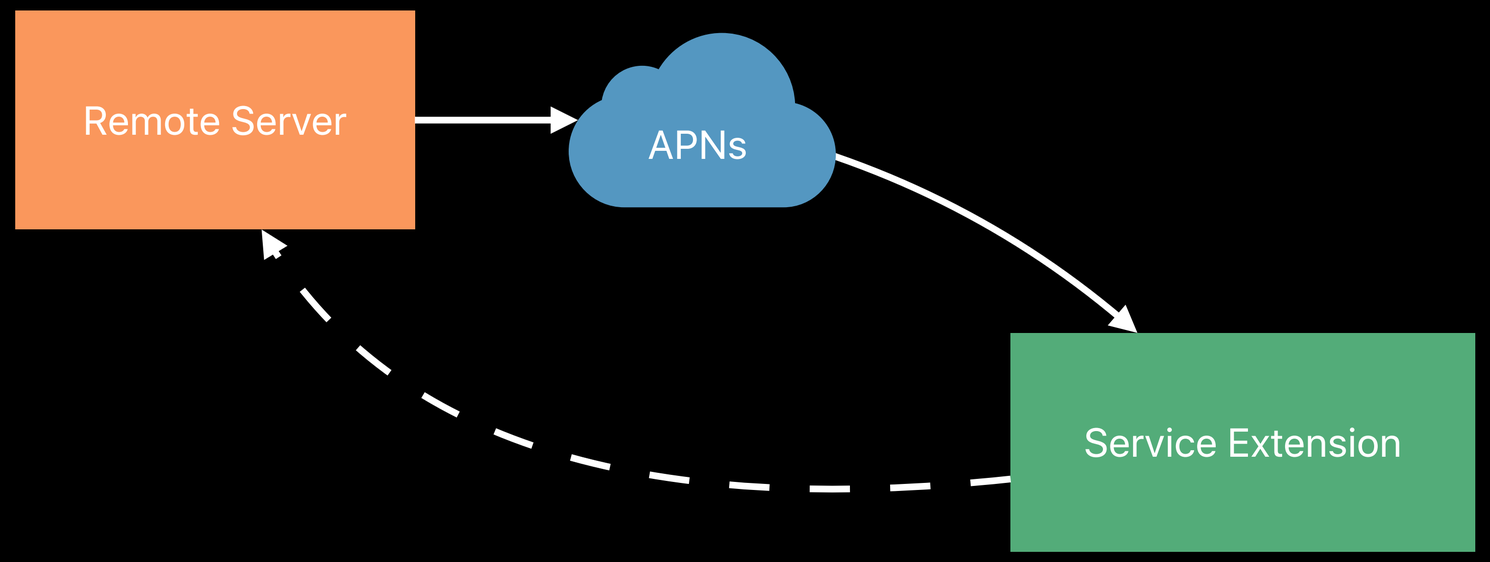
Modifying Push Content

Example



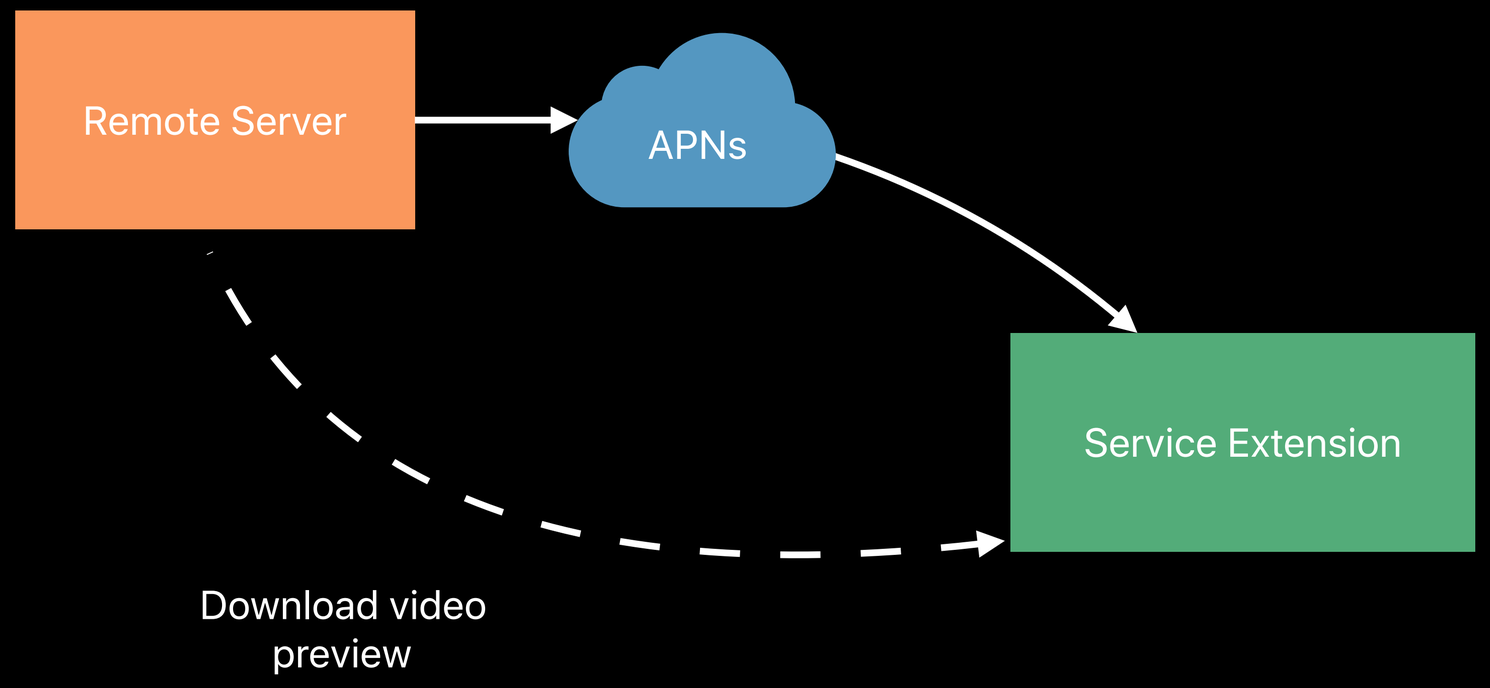
Modifying Push Content

Example



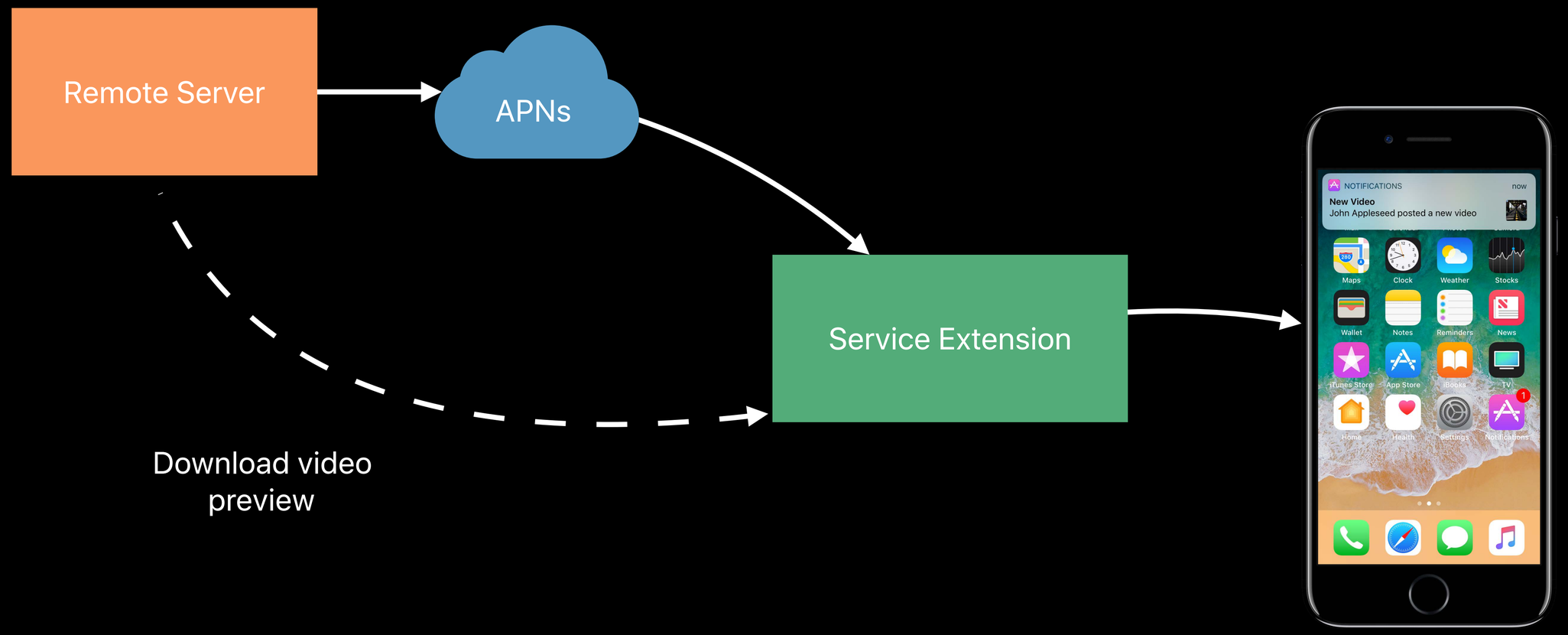
Modifying Push Content

Example



Modifying Push Content

Example



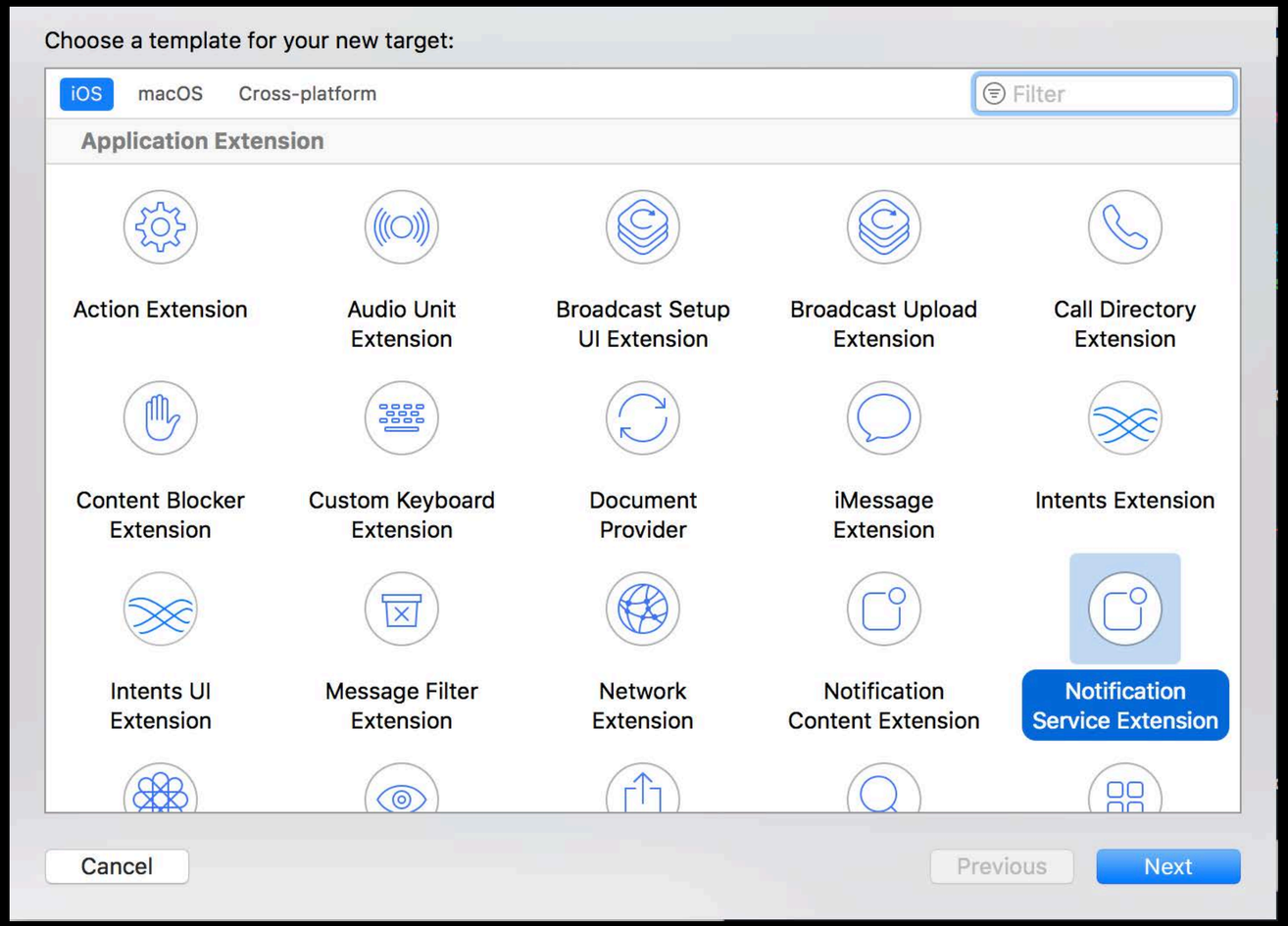
```
// Modifying Push Content

{
    "aps": {
        "alert": { … },
        "mutable-content": 1
    },
    "type": "video",
    "video-url": "https://www.example.com/media/myvideo.m4a",
    "video-location": "Skate Park, SF",
    "video-user": "John Appleseed"
}
```

```
// Modifying Push Content

{
    "aps": {
        "alert": { … },
        "mutable-content": 1
    },
    "type": "video",
    "video-url": "https://www.example.com/media/myvideo.m4a",
    "video-location": "Skate Park, SF",
    "video-user": "John Appleseed"
}
```

```
// Modifying Push Content

{
    "aps": {
        "alert": { … },
        "mutable-content": 1
    },
    "type": "video",
    "video-url": "https://www.example.com/media/myvideo.m4a",
    "video-location": "Skate Park, SF",
    "video-user": "John Appleseed"
}
```

Modifying Push Content

Service Extension Template



```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {

    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {

        self.contentHandler = contentHandler
        self.bestAttemptContent = (request.content.mutableCopy()
            as? UNMutableNotificationContent)

        … // Handle request before presenting
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {

    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {

        self.contentHandler = contentHandler
        self.bestAttemptContent = (request.content.mutableCopy()
            as? UNMutableNotificationContent)

        … // Handle request before presenting
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {

    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {

        self.contentHandler = contentHandler
        self.bestAttemptContent = (request.content.mutableCopy()
            as? UNMutableNotificationContent)

        … // Handle request before presenting
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {

    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {

        self.contentHandler = contentHandler
        self.bestAttemptContent = (request.content.mutableCopy()
            as? UNMutableNotificationContent)

        … // Handle request before presenting
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {

    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {

        self.contentHandler = contentHandler
        self.bestAttemptContent = (request.content.mutableCopy()
            as? UNMutableNotificationContent)

        … // Handle request before presenting
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {

    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {

        self.contentHandler = contentHandler
        self.bestAttemptContent = (request.content.mutableCopy()
            as? UNMutableNotificationContent)

        … // Handle request before presenting
    }
}
```

```swift
// Modifying Push Content

override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {
    …
    if let mutableContent = self.bestAttemptContent,
    let urlString = mutableContent.userInfo["video-url"] as? String {

        self.download(url: urlString) { (fileLocation) in
            if let attachment = try? UNNotificationAttachment(identifier: "video",
                                    url: fileLocation) {
                mutableContent.attachments = [ attachment ]
                contentHandler(mutableContent)
            }
        }
    }
 }
```

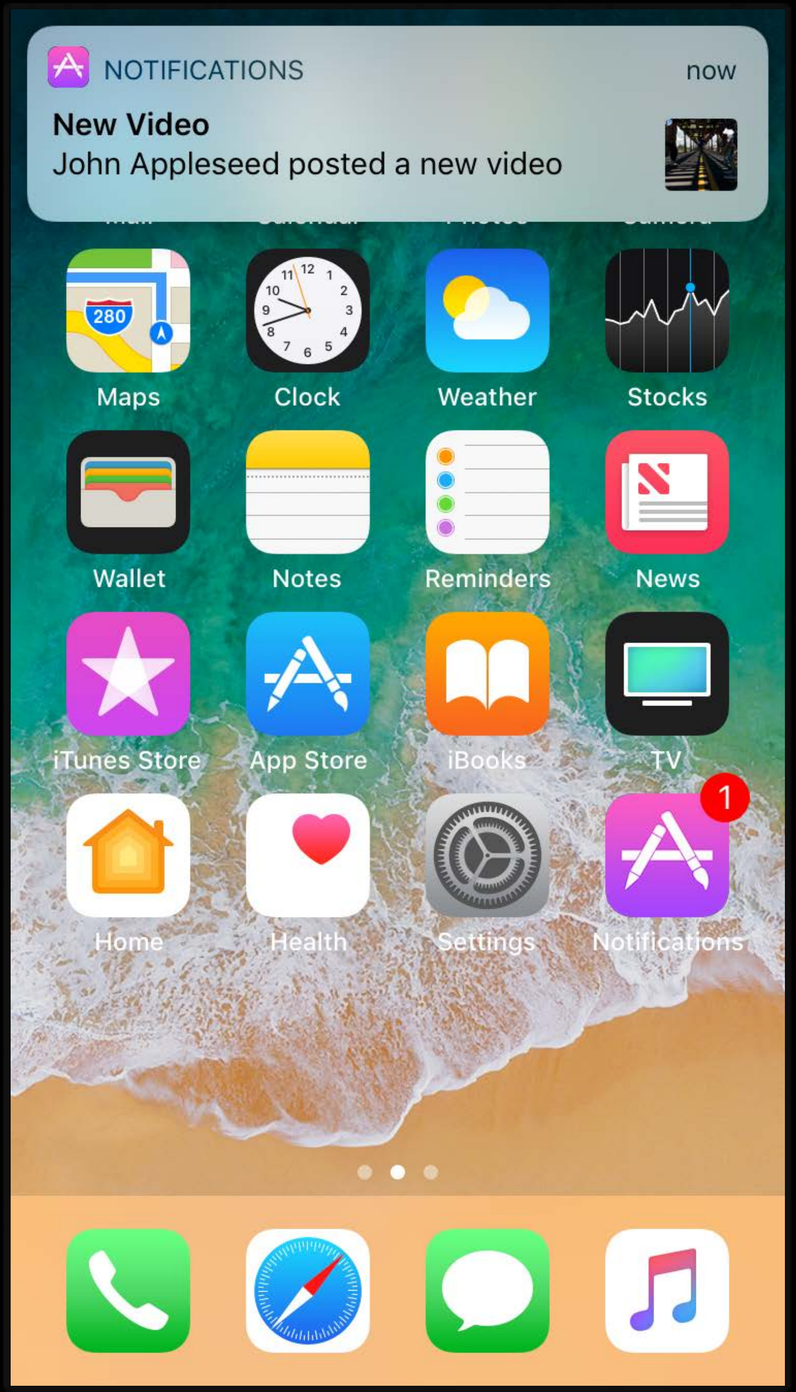
```
// Modifying Push Content

override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {
    …
    if let mutableContent = self.bestAttemptContent,
    let urlString = mutableContent.userInfo["video-url"] as? String {

        self.download(url: urlString) { (fileLocation) in
            if let attachment = try? UNNotificationAttachment(identifier: "video",
                                    url: fileLocation) {
                mutableContent.attachments = [ attachment ]
                contentHandler(mutableContent)
            }
        }
    }
 }
```

```
// Modifying Push Content

override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {
    …
    if let mutableContent = self.bestAttemptContent,
    let urlString = mutableContent.userInfo["video-url"] as? String {

        self.download(url: urlString) { (fileLocation) in
            if let attachment = try? UNNotificationAttachment(identifier: "video",
                                    url: fileLocation) {
                mutableContent.attachments = [ attachment ]
                contentHandler(mutableContent)
            }
        }
    }
 }
```

```swift
// Modifying Push Content

override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {
    …
    if let mutableContent = self.bestAttemptContent,
    let urlString = mutableContent.userInfo["video-url"] as? String {

        self.download(url: urlString) { (fileLocation) in
            if let attachment = try? UNNotificationAttachment(identifier: "video",
                                    url: fileLocation) {
                mutableContent.attachments = [ attachment ]
                contentHandler(mutableContent)
            }
        }
    }
 }
```

```swift
// Modifying Push Content

override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {
    …
    if let mutableContent = self.bestAttemptContent,
    let urlString = mutableContent.userInfo["video-url"] as? String {

        self.download(url: urlString) { (fileLocation) in
            if let attachment = try? UNNotificationAttachment(identifier: "video",
                                    url: fileLocation) {
                mutableContent.attachments = [ attachment ]
                contentHandler(mutableContent)
            }
        }
    }
}
```

```swift
// Modifying Push Content

override public func didReceive(_ request: UNNotificationRequest, @escaping
        withContentHandler contentHandler: (UNNotificationContent) -> Void) {
    …
    if let mutableContent = self.bestAttemptContent,
    let urlString = mutableContent.userInfo["video-url"] as? String {

        self.download(url: urlString) { (fileLocation) in
            if let attachment = try? UNNotificationAttachment(identifier: "video",
                                    url: fileLocation) {
                mutableContent.attachments = [ attachment ]
                contentHandler(mutableContent)
            }
        }
    }
 }
```


NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {
    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override func serviceExtensionTimeWillExpire() {

        if let contentHandler = self.contentHandler,
           let bestAttemptContent = self.bestAttemptContent,
           let mediaType = bestAttemptContent.userInfo["type"] as? String {
                bestAttemptContent.title = "Incoming \(mediaType)!",

                contentHandler(bestAttemptContent)
        }
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {
    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override func serviceExtensionTimeWillExpire() {

        if let contentHandler = self.contentHandler,
           let bestAttemptContent = self.bestAttemptContent,
           let mediaType = bestAttemptContent.userInfo["type"] as? String {
                bestAttemptContent.title = "Incoming \(mediaType)!",

                contentHandler(bestAttemptContent)
        }
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {
    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override func serviceExtensionTimeWillExpire() {

        if let contentHandler = self.contentHandler,
           let bestAttemptContent = self.bestAttemptContent,
           let mediaType = bestAttemptContent.userInfo["type"] as? String {
                bestAttemptContent.title = "Incoming \(mediaType)!",

                contentHandler(bestAttemptContent)
        }
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {
    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override func serviceExtensionTimeWillExpire() {

        if let contentHandler = self.contentHandler,
           let bestAttemptContent = self.bestAttemptContent,
           let mediaType = bestAttemptContent.userInfo["type"] as? String {
                bestAttemptContent.title = "Incoming \(mediaType)!",

                contentHandler(bestAttemptContent)
        }
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {
    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override func serviceExtensionTimeWillExpire() {

        if let contentHandler = self.contentHandler,
           let bestAttemptContent = self.bestAttemptContent,
           let mediaType = bestAttemptContent.userInfo["type"] as? String {
                bestAttemptContent.title = "Incoming \(mediaType)!",

                contentHandler(bestAttemptContent)
        }
    }
}
```

```swift
// Modifying Push Content

public class NotificationService: UNNotificationServiceExtension {
    var contentHandler: ((UNNotificationContent) -> Void)?
    var bestAttemptContent: UNMutableNotificationContent?

    override func serviceExtensionTimeWillExpire() {

        if let contentHandler = self.contentHandler,
           let bestAttemptContent = self.bestAttemptContent,
           let mediaType = bestAttemptContent.userInfo["type"] as? String {
                bestAttemptContent.title = "Incoming \(mediaType)!",

                contentHandler(bestAttemptContent)
        }
    }
}
```

Modifying Push Content

Service Extension best practices

Modifying Push Content

Service Extension best practices

Modifying Push Content

Service Extension best practices

Short execution time and memory limit

Modifying Push Content

Service Extension best practices

Short execution time and memory limit

Fallback

Modifying Push Content

Service Extension best practices

Short execution time and memory limit

Fallback

Notifications will be presented

Modifying Push Content

Service Extension best practices

Short execution time and memory limit

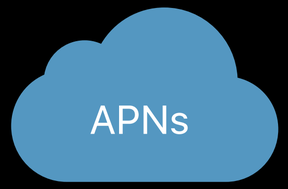
Fallback

Notifications will be presented

Different from silent notifications

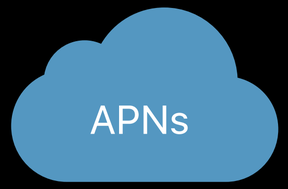
Modifying Push Content

Service Extension best practices



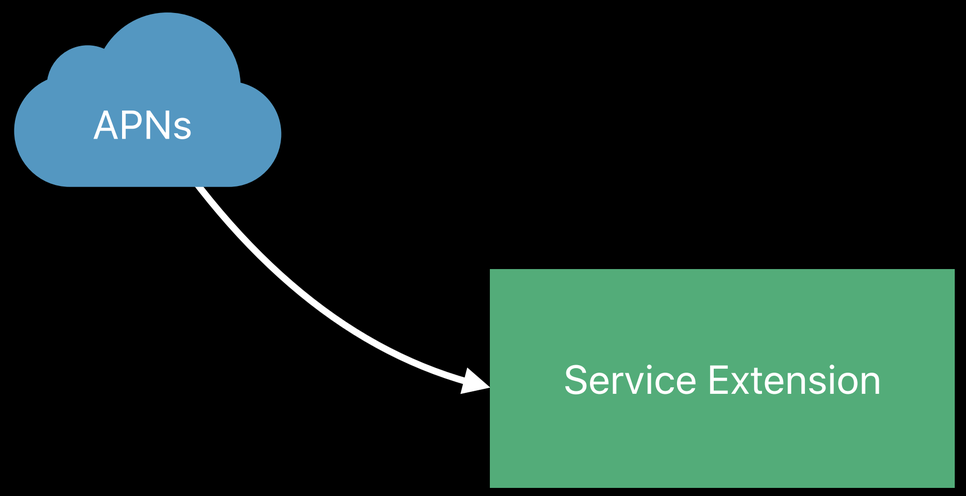
Modifying Push Content

Service Extension best practices



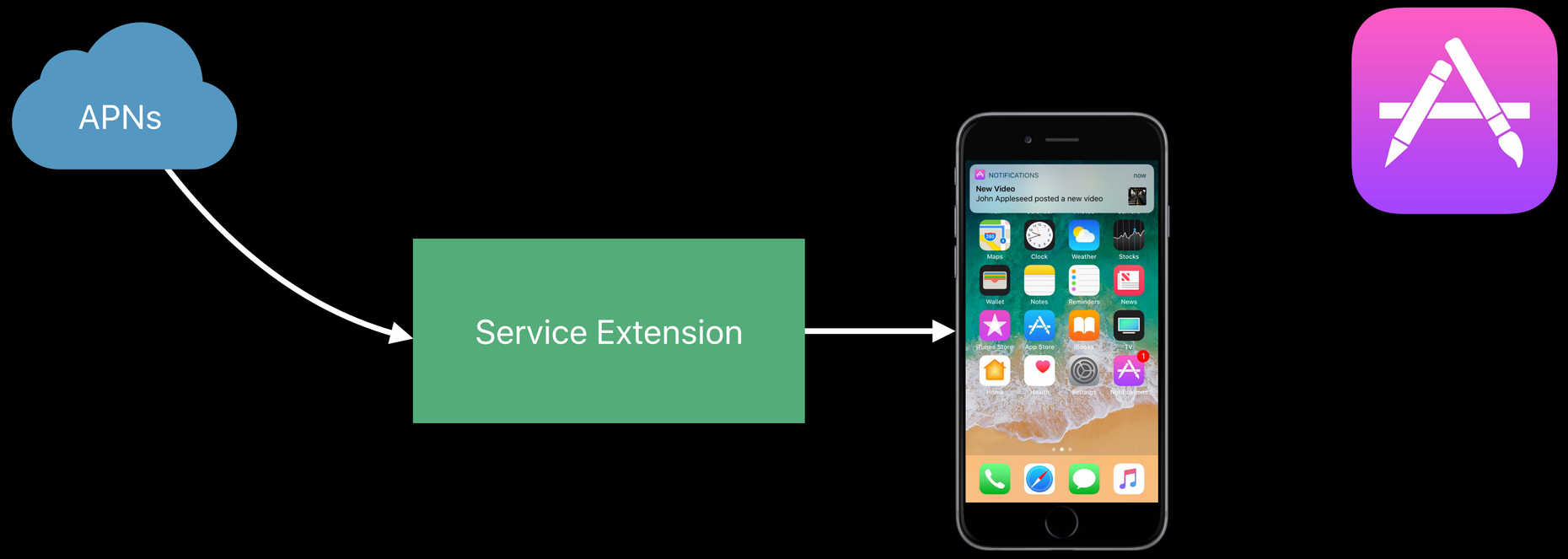
Modifying Push Content

Service Extension best practices



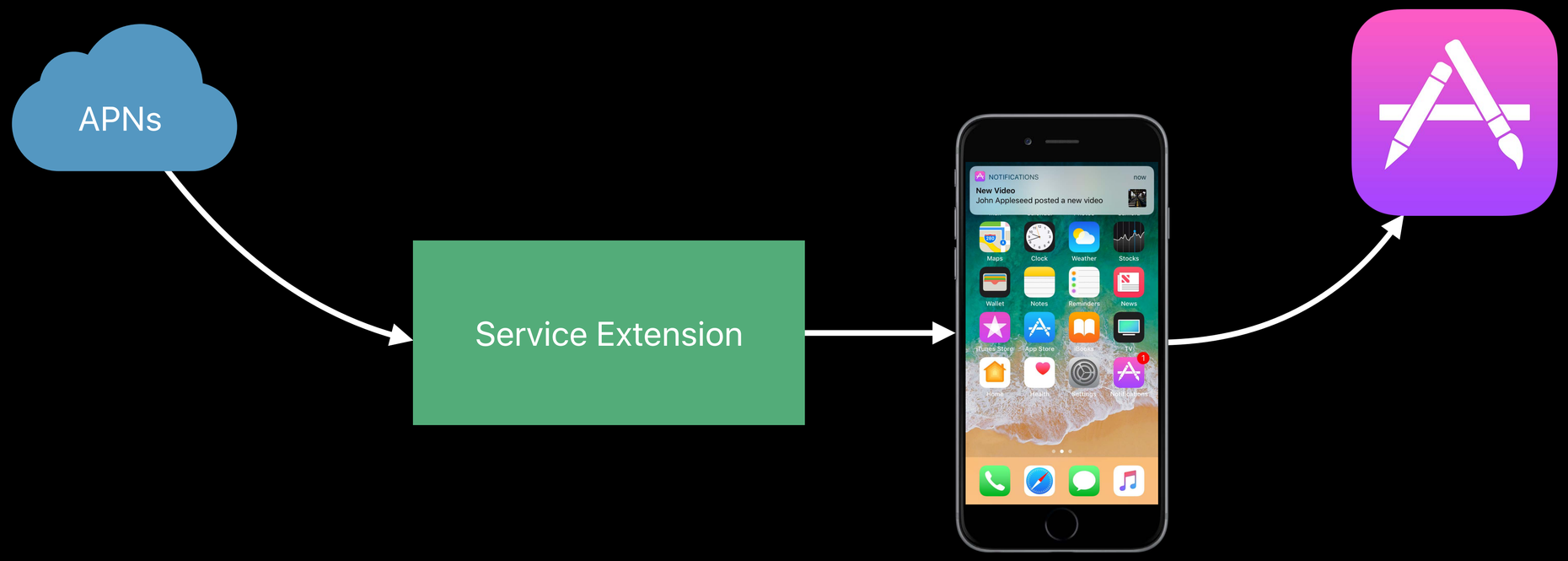
Modifying Push Content

Service Extension best practices



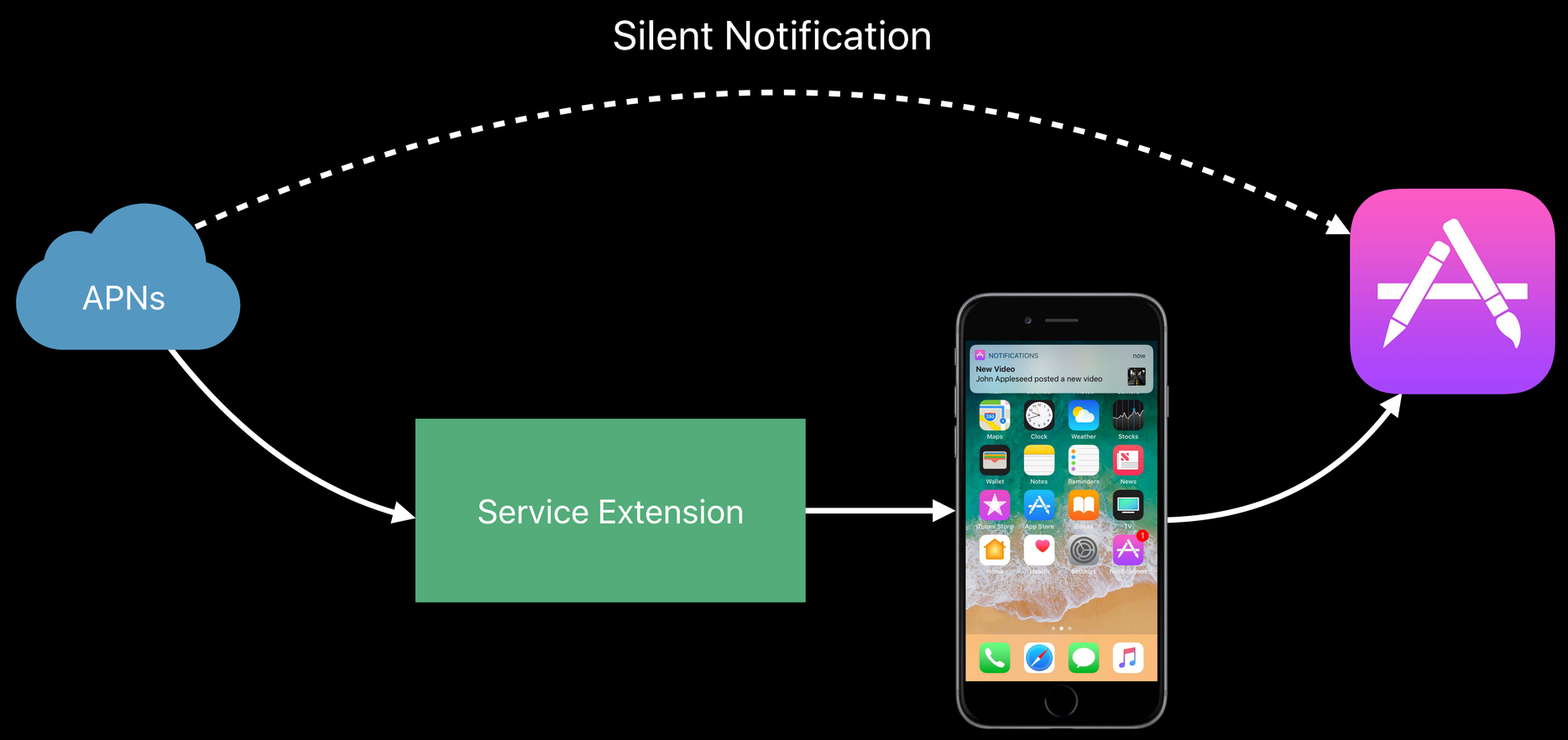
Modifying Push Content

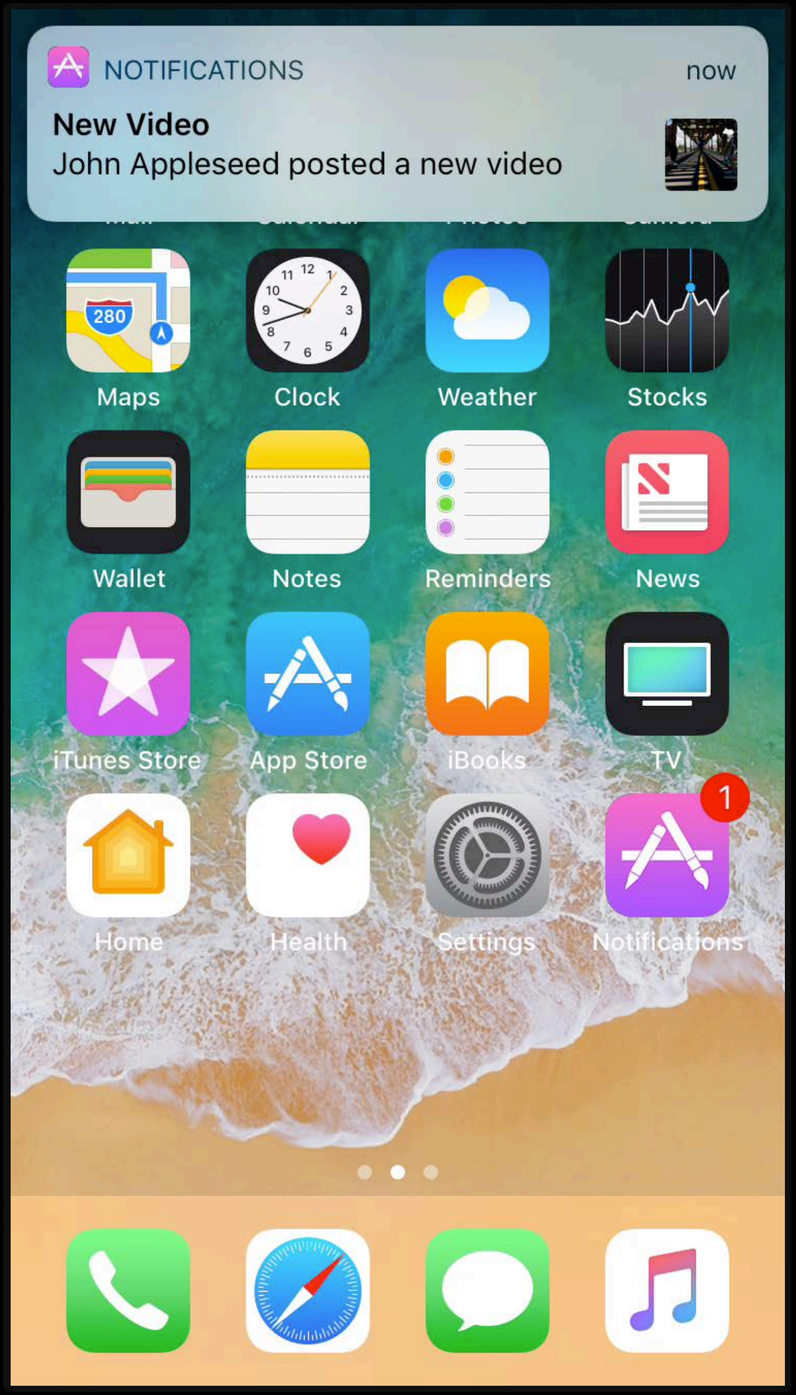
Service Extension best practices

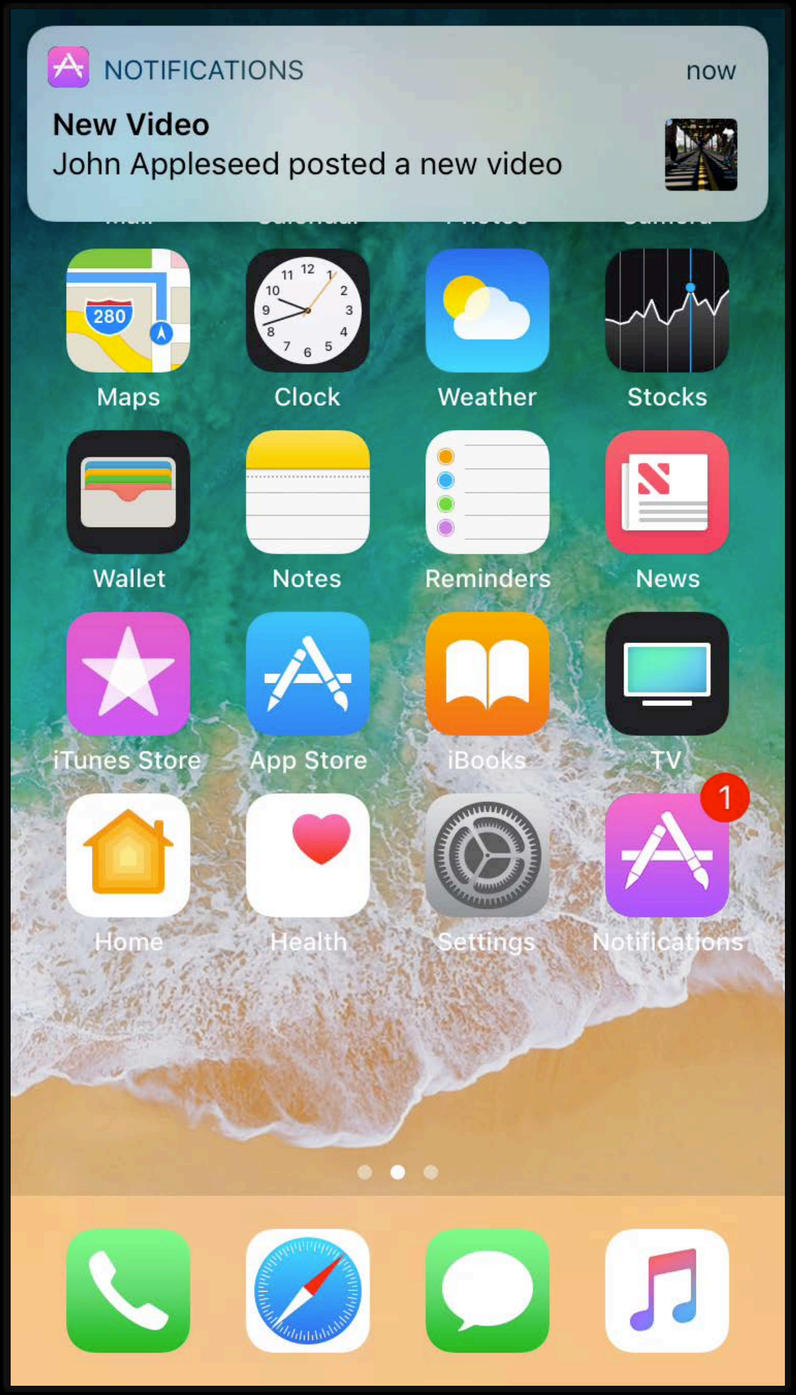


Modifying Push Content

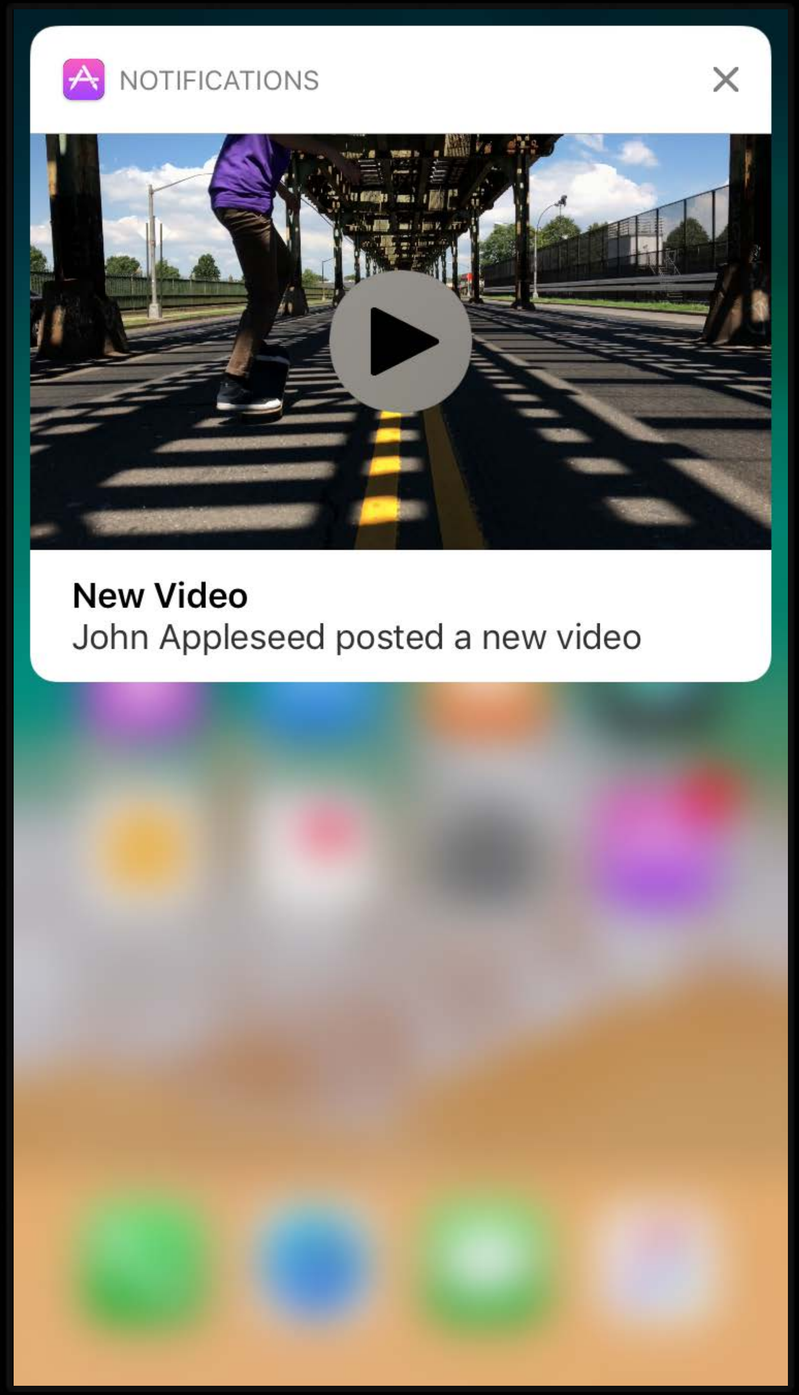
Service Extension best practices




NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1


NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1

Customizing Rich Notifications


NOTIFICATIONS
New Video
John Appleseed posted a new video

Customizing Rich Notifications

Motivation

Customizing Rich Notifications

Motivation

App specific look and feel

Customizing Rich Notifications

Motivation

App specific look and feel

Custom display of notification content

Customizing Rich Notifications

Motivation

App specific look and feel

Custom display of notification content

Interactive and dynamic UI

Content Extensions

Customizing Rich Notifications

Content Extension

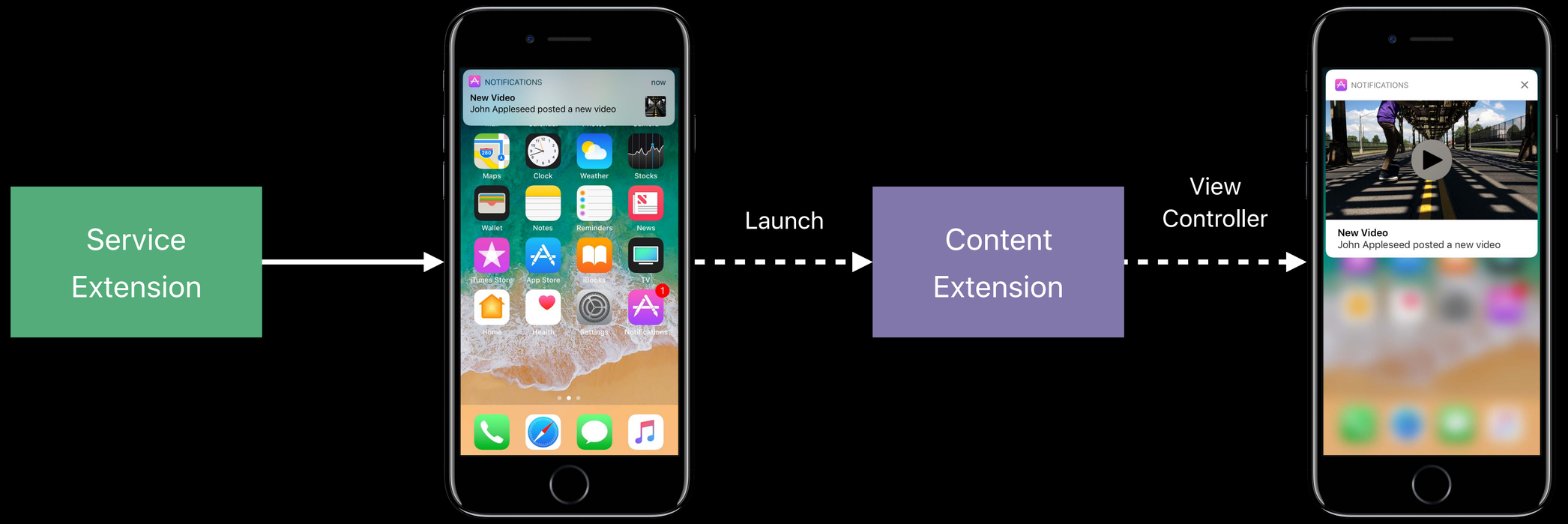
Service Extension

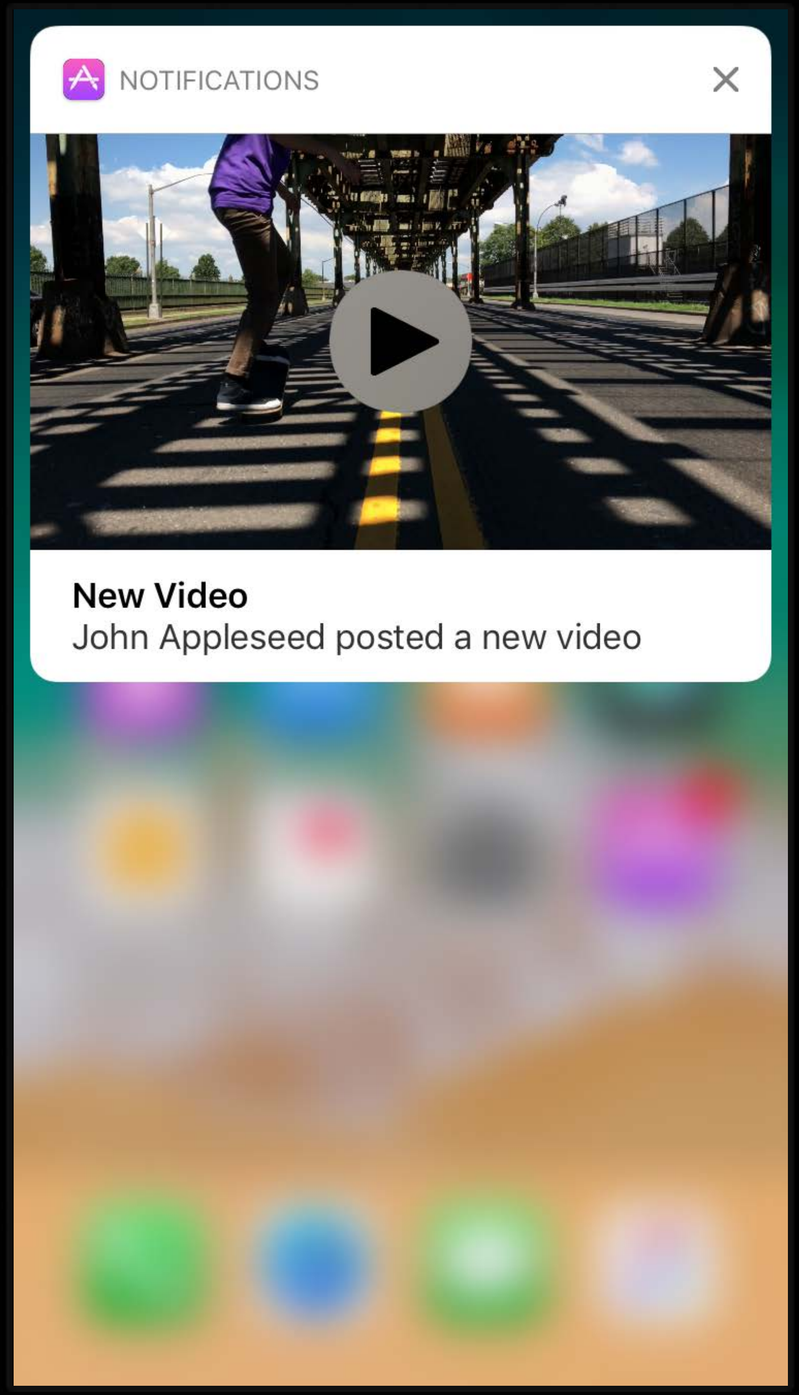
Customizing Rich Notifications

Content Extension

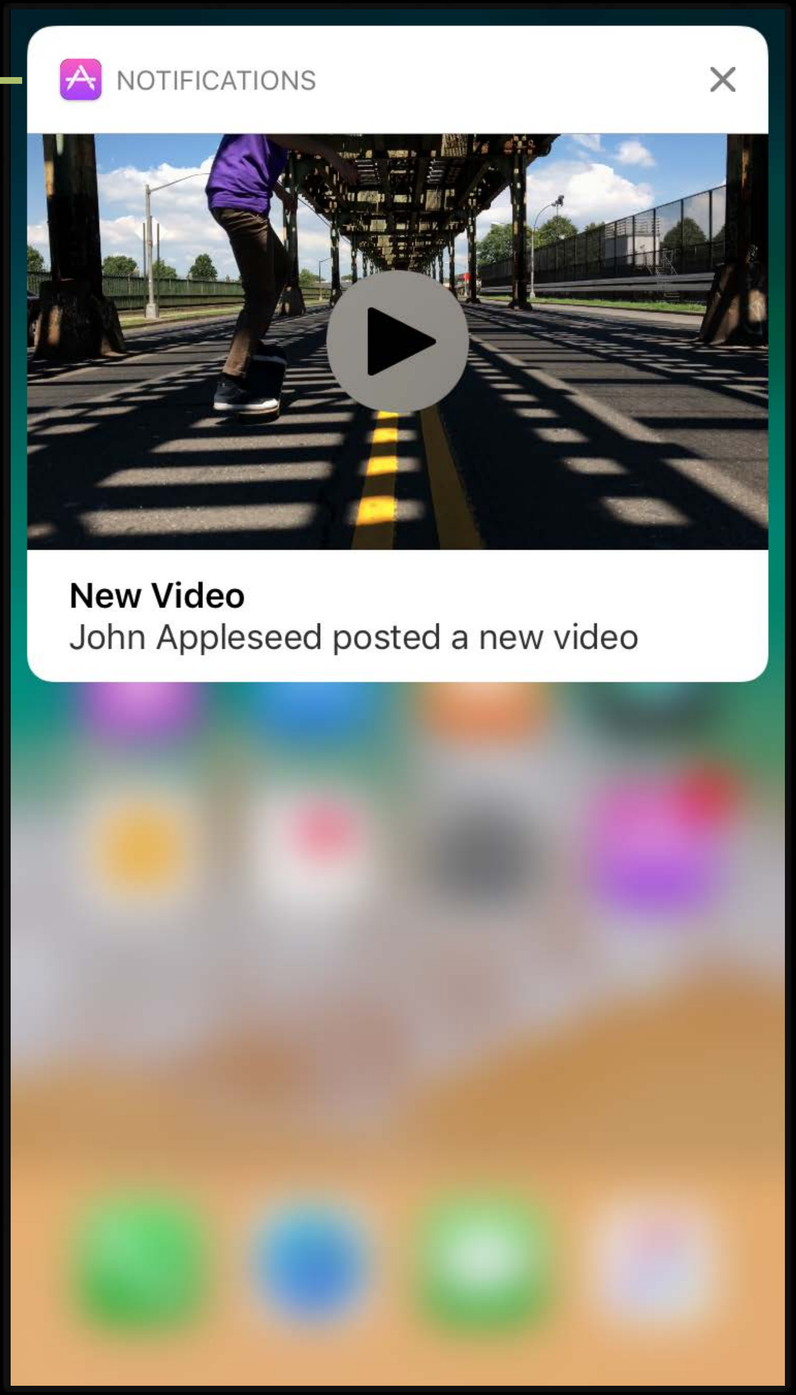
Customizing Rich Notifications

Content Extension



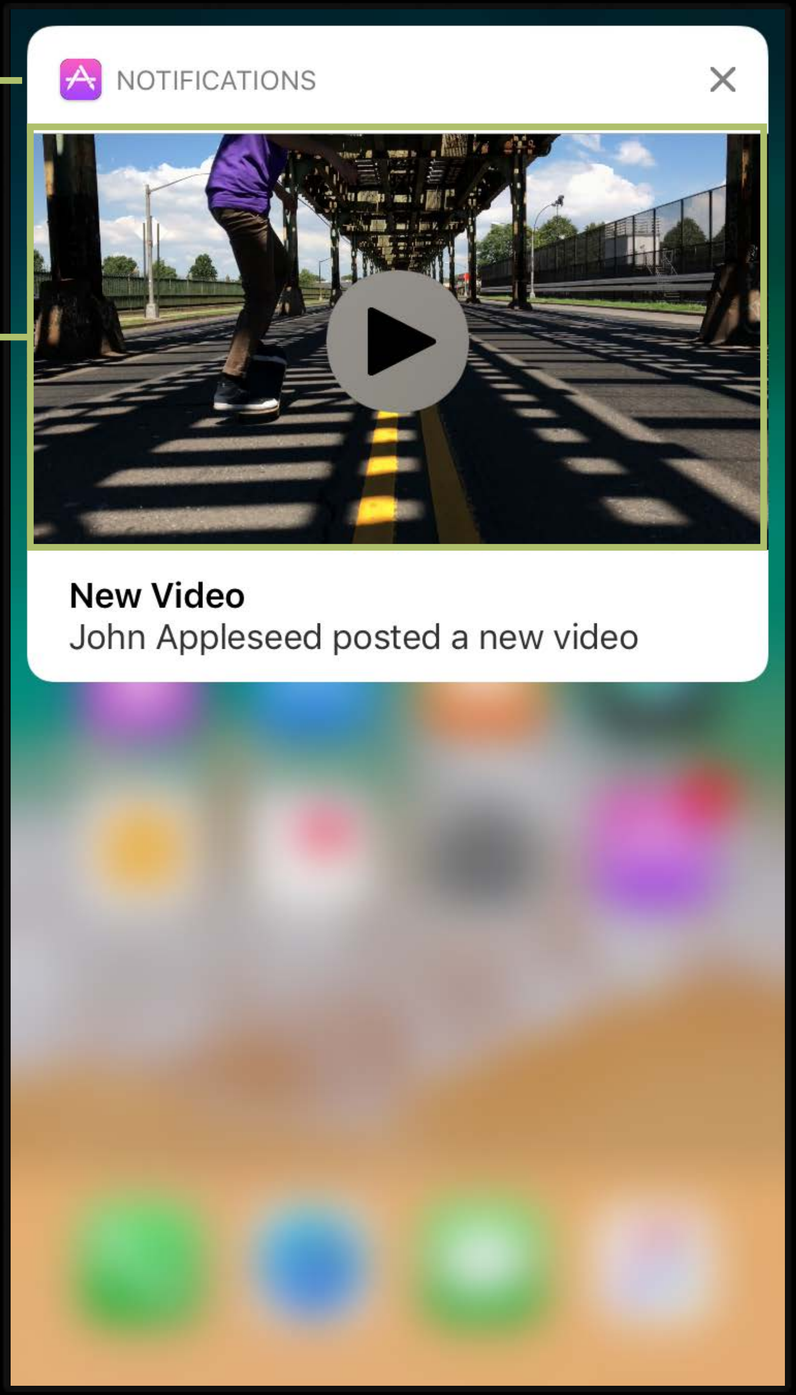

NOTIFICATIONS
New Video
John Appleseed posted a new video

Customize Title



Customize Title

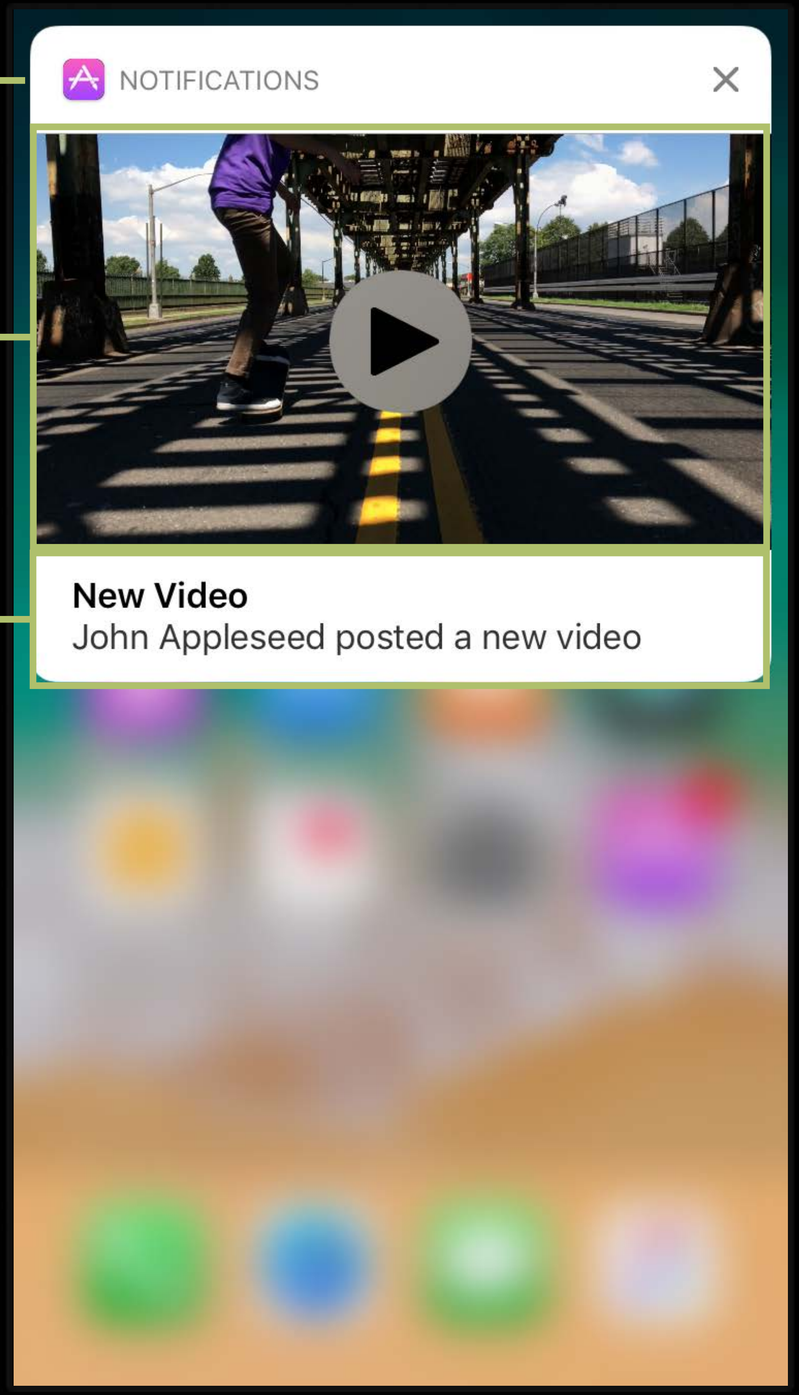
Customize UI



Customize Title

Customize UI

Remove
Default Content

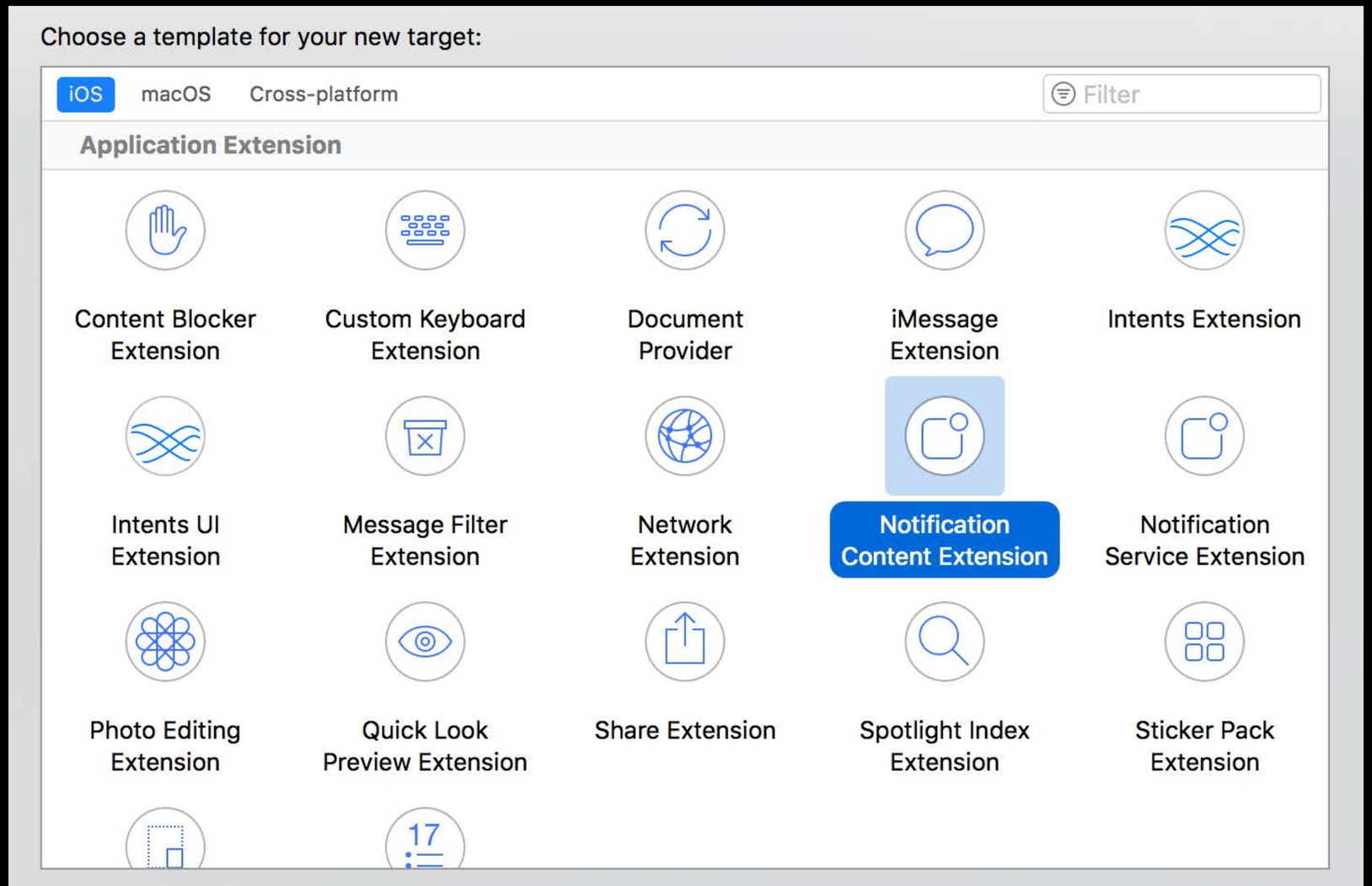


Customizing Rich Notifications

Content Extension Template

Customizing Rich Notifications

Content Extension Template



Customizing Rich Notifications

Content Extension `Info.plist`

Customizing Rich Notifications

Content Extension `Info.plist`

Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	NotificationMediaAttachmentExtension
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(4 items)
UNNotificationExtensionCategory	String	video-playback-category
UNNotificationExtensionDefaultContentHidden	Boolean	YES
UNNotificationExtensionOverridesDefaultTitle	Boolean	YES
UNNotificationExtensionInitialContentSizeRatio	Number	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension

Customizing Rich Notifications

Content Extension `Info.plist`

Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	NotificationMediaAttachmentExtension
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(4 items)
UNNotificationExtensionCategory	String	video-playback-category
UNNotificationExtensionDefaultContentHidden	Boolean	YES
UNNotificationExtensionOverridesDefaultTitle	Boolean	YES
UNNotificationExtensionInitialContentSizeRatio	Number	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension

Customizing Rich Notifications

Content Extension `Info.plist`

Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	NotificationMediaAttachmentExtension
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(4 items)
UNNotificationExtensionCategory	String	video-playback-category
UNNotificationExtensionDefaultContentHidden	Boolean	YES
UNNotificationExtensionOverridesDefaultTitle	Boolean	YES
UNNotificationExtensionInitialContentSizeRatio	Number	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension

Customizing Rich Notifications

Content Extension `Info.plist`

Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	NotificationMediaAttachmentExtension
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(4 items)
UNNotificationExtensionCategory	String	video-playback-category
UNNotificationExtensionDefaultContentHidden	Boolean	YES
UNNotificationExtensionOverridesDefaultTitle	Boolean	YES
UNNotificationExtensionInitialContentSizeRatio	Number	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension

```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var videoPlayer: AVPlayer?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        if let attachment = content.attachments.first {
            if attachment.url.startAccessingSecurityScopedResource() {
                videoPlayer = AVPlayer(url: attachment.url)
                attachment.url.stopAccessingSecurityScopedResource()
            }
        }
    }
}
```

```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var videoPlayer: AVPlayer?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        if let attachment = content.attachments.first {
            if attachment.url.startAccessingSecurityScopedResource() {
                videoPlayer = AVPlayer(url: attachment.url)
                attachment.url.stopAccessingSecurityScopedResource()
            }
        }
    }
}
```

```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var videoPlayer: AVPlayer?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        if let attachment = content.attachments.first {
            if attachment.url.startAccessingSecurityScopedResource() {
                videoPlayer = AVPlayer(url: attachment.url)
                attachment.url.stopAccessingSecurityScopedResource()
            }
        }
    }
}
```

```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var videoPlayer: AVPlayer?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        if let attachment = content.attachments.first {
            if attachment.url.startAccessingSecurityScopedResource() {
                videoPlayer = AVPlayer(url: attachment.url)
                attachment.url.stopAccessingSecurityScopedResource()
            }
        }
    }
}
```

```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var videoPlayer: AVPlayer?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        if let attachment = content.attachments.first {
            if attachment.url.startAccessingSecurityScopedResource() {
                videoPlayer = AVPlayer(url: attachment.url)
                attachment.url.stopAccessingSecurityScopedResource()
            }
        }
    }
}
```

```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var videoPlayer: AVPlayer?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        if let attachment = content.attachments.first {
            if attachment.url.startAccessingSecurityScopedResource() {
                videoPlayer = AVPlayer(url: attachment.url)
                attachment.url.stopAccessingSecurityScopedResource()
            }
        }
    }
}
```

```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var userLabel: UILabel?
    @IBOutlet var locationLabel: UILabel?
    @IBOutlet var descriptionLabel: UILabel?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        self.title = content.title

        let userInfo = content.userInfo
        self.userLabel?.text = userInfo["video-user"] as? String
        self.locationLabel?.text = userInfo["video-location"] as? String
        self.descriptionLabel?.text = userInfo["video-description"] as? String
    }
}
```

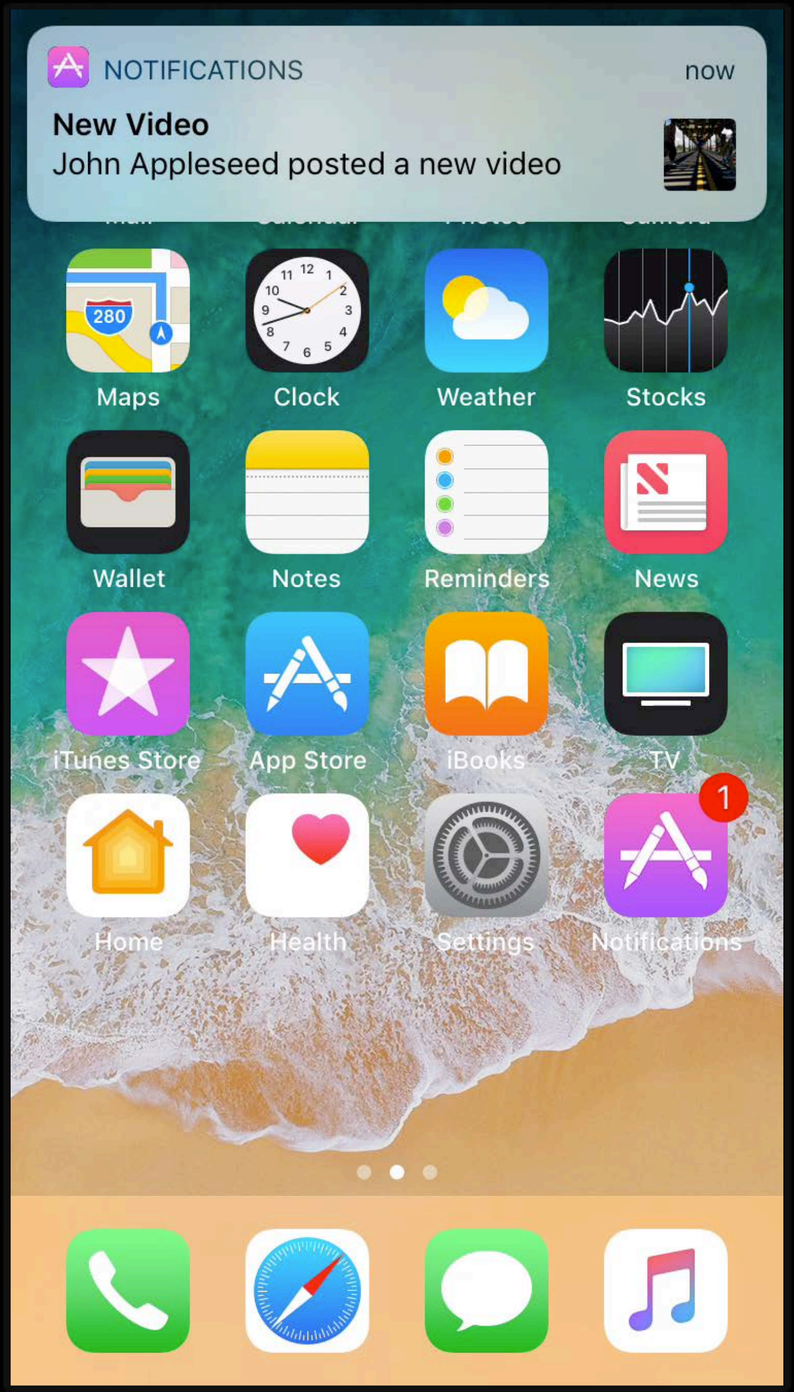
```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var userLabel: UILabel?
    @IBOutlet var locationLabel: UILabel?
    @IBOutlet var descriptionLabel: UILabel?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        self.title = content.title

        let userInfo = content.userInfo
        self.userLabel?.text = userInfo["video-user"] as? String
        self.locationLabel?.text = userInfo["video-location"] as? String
        self.descriptionLabel?.text = userInfo["video-description"] as? String
    }
}
```

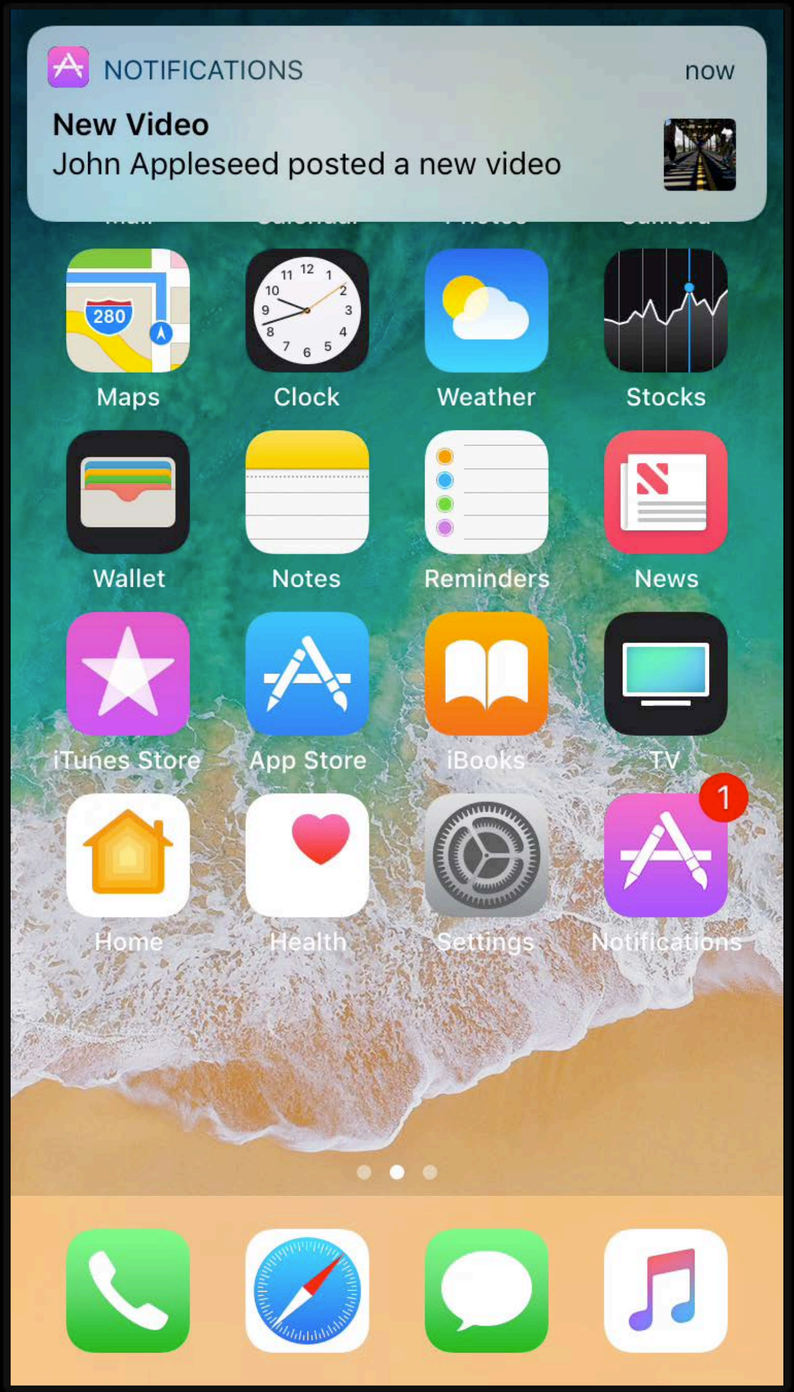
```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var userLabel: UILabel?
    @IBOutlet var locationLabel: UILabel?
    @IBOutlet var descriptionLabel: UILabel?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        self.title = content.title

        let userInfo = content.userInfo
        self.userLabel?.text = userInfo["video-user"] as? String
        self.locationLabel?.text = userInfo["video-location"] as? String
        self.descriptionLabel?.text = userInfo["video-description"] as? String
    }
}
```

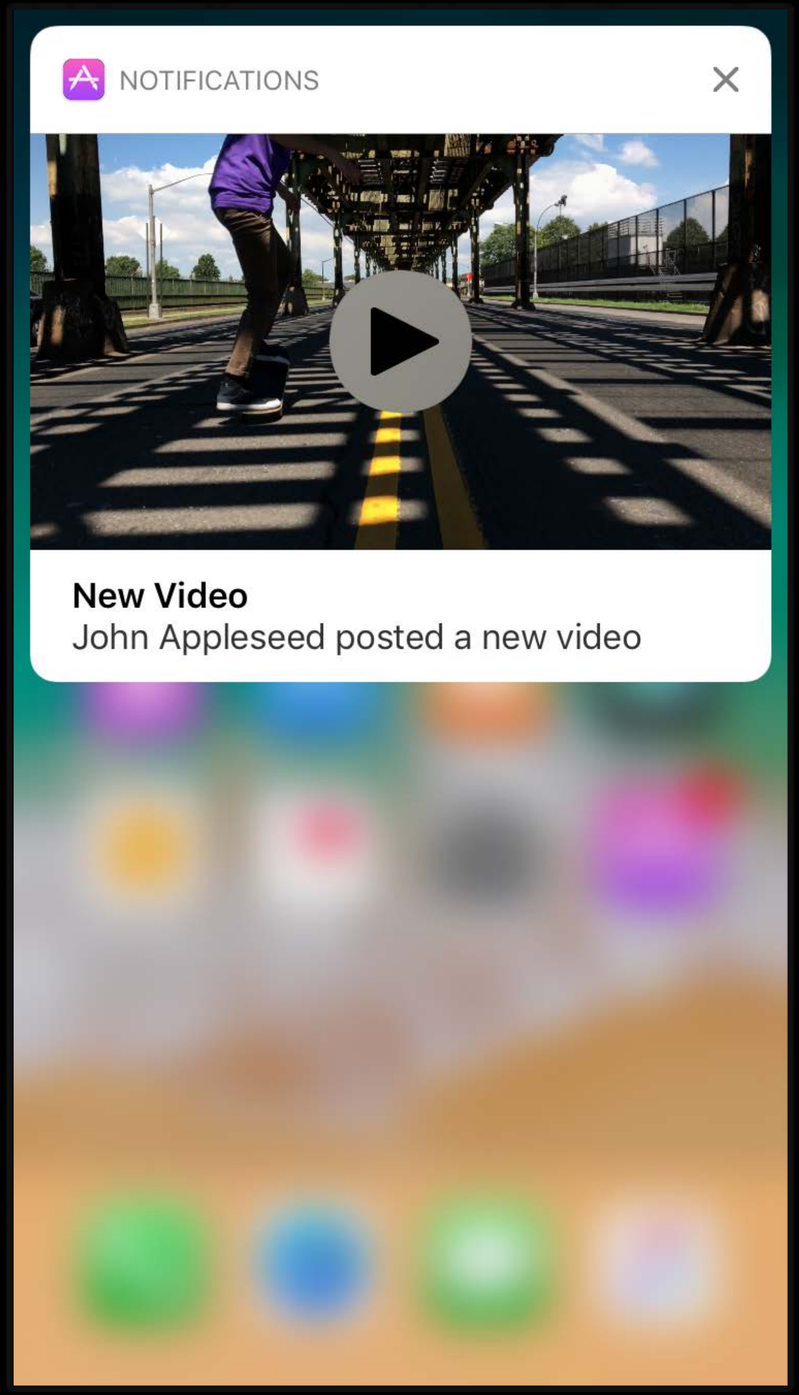
```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var userLabel: UILabel?
    @IBOutlet var locationLabel: UILabel?
    @IBOutlet var descriptionLabel: UILabel?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        self.title = content.title

        let userInfo = content.userInfo
        self.userLabel?.text = userInfo["video-user"] as? String
        self.locationLabel?.text = userInfo["video-location"] as? String
        self.descriptionLabel?.text = userInfo["video-description"] as? String
    }
}
```

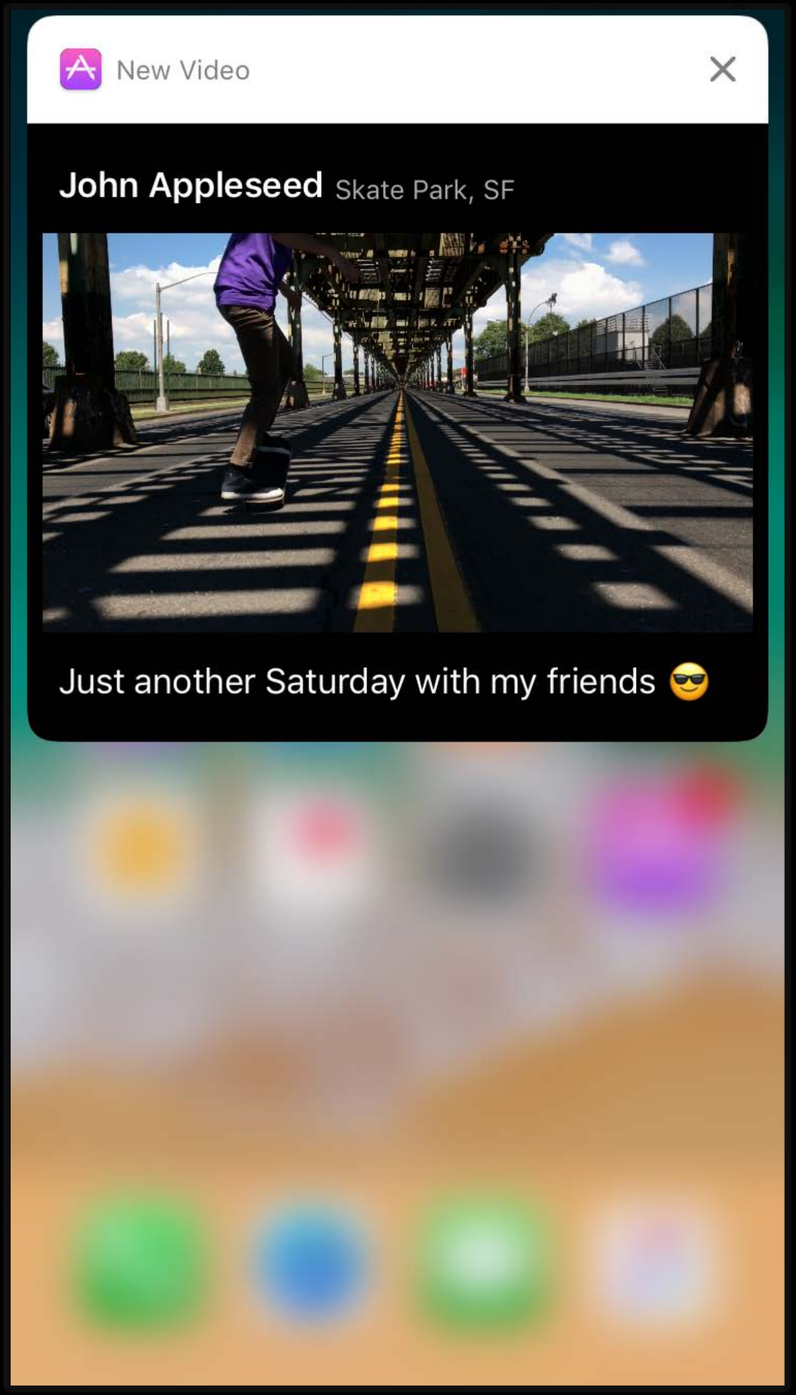
```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var userLabel: UILabel?
    @IBOutlet var locationLabel: UILabel?
    @IBOutlet var descriptionLabel: UILabel?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        self.title = content.title

        let userInfo = content.userInfo
        self.userLabel?.text = userInfo["video-user"] as? String
        self.locationLabel?.text = userInfo["video-location"] as? String
        self.descriptionLabel?.text = userInfo["video-description"] as? String
    }
}
```

```swift
// Content Extension Custom UI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var userLabel: UILabel?
    @IBOutlet var locationLabel: UILabel?
    @IBOutlet var descriptionLabel: UILabel?

    func didReceive(_ notification: UNNotification) {
        let content = notification.request.content
        self.title = content.title

        let userInfo = content.userInfo
        self.userLabel?.text = userInfo["video-user"] as? String
        self.locationLabel?.text = userInfo["video-location"] as? String
        self.descriptionLabel?.text = userInfo["video-description"] as? String
    }
}
```

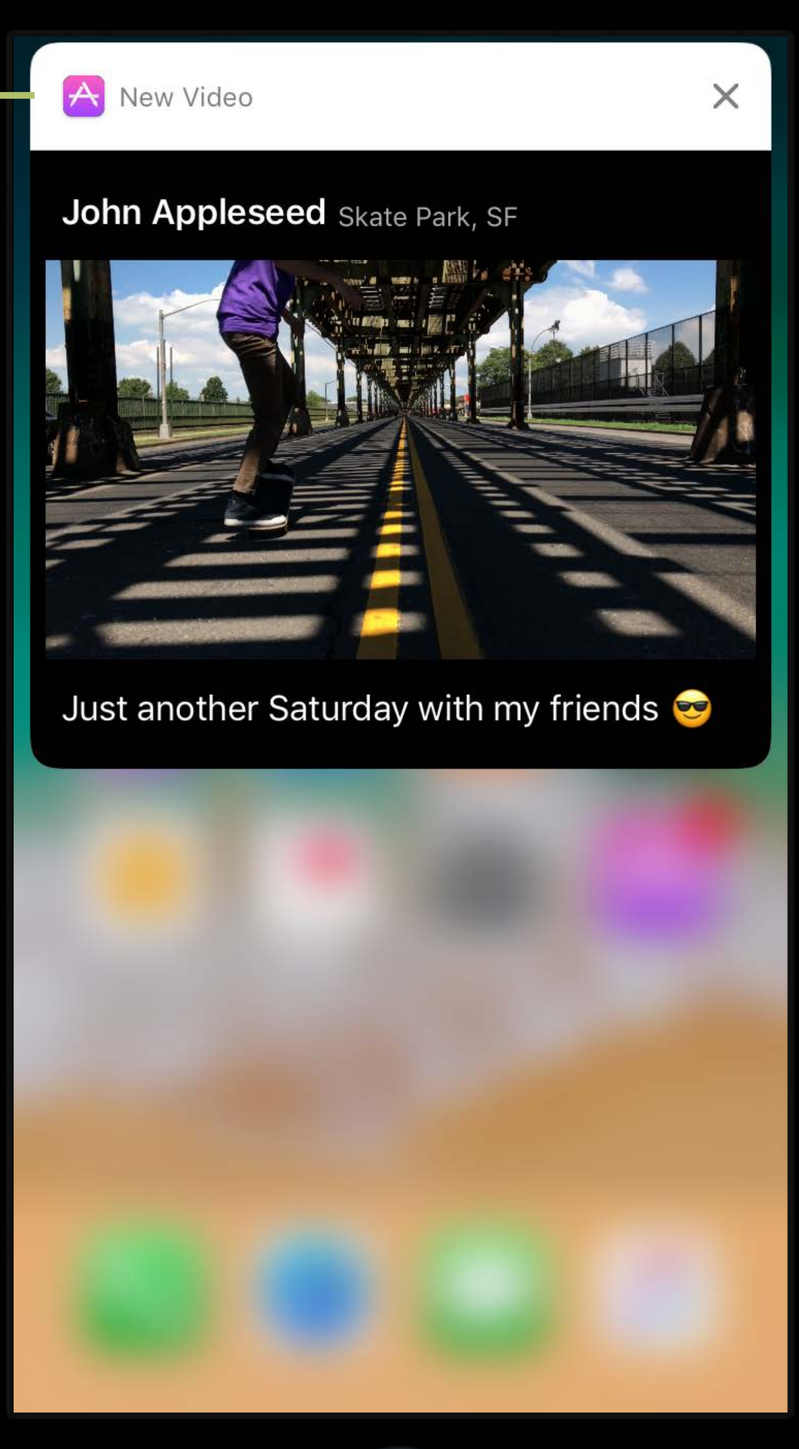

NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1


NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1


NOTIFICATIONS
New Video
John Appleseed posted a new video

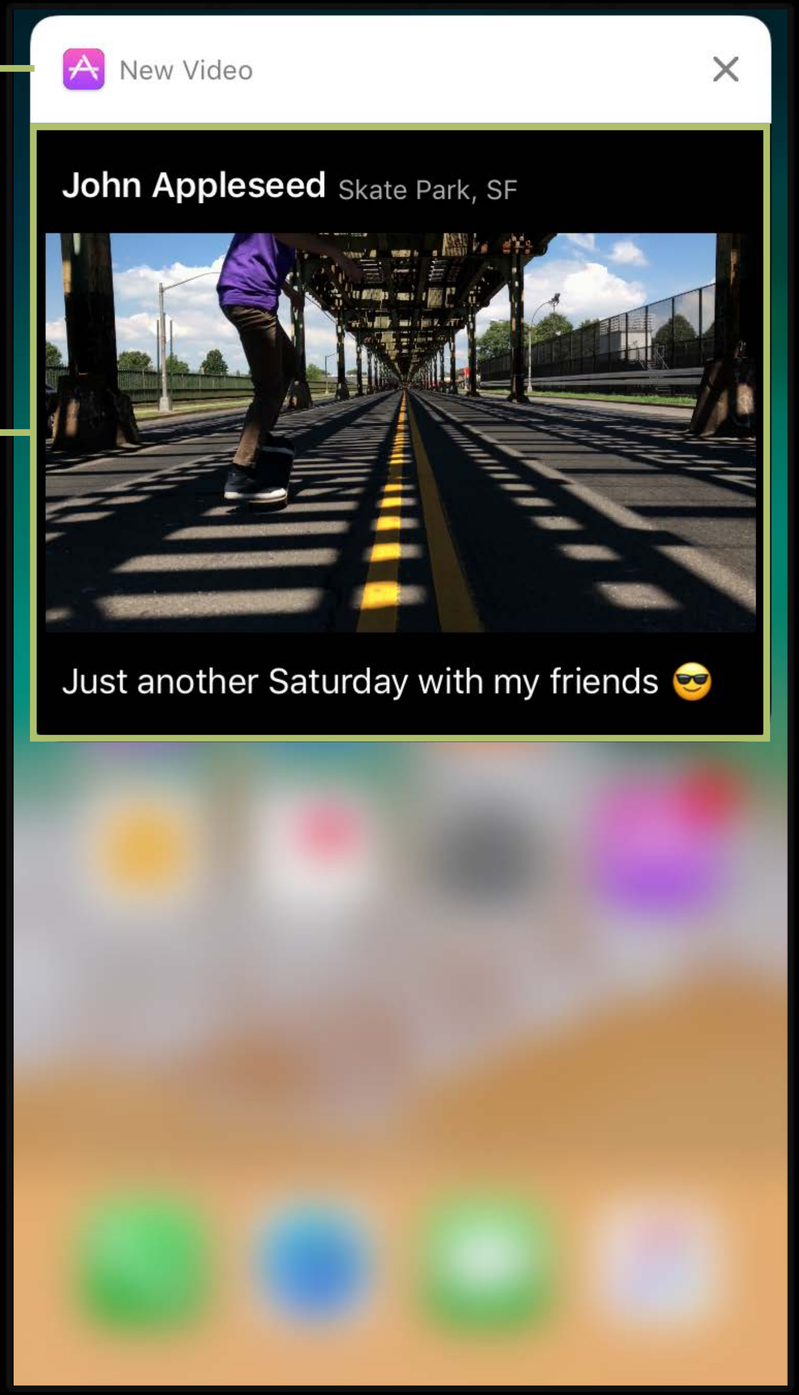

New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎

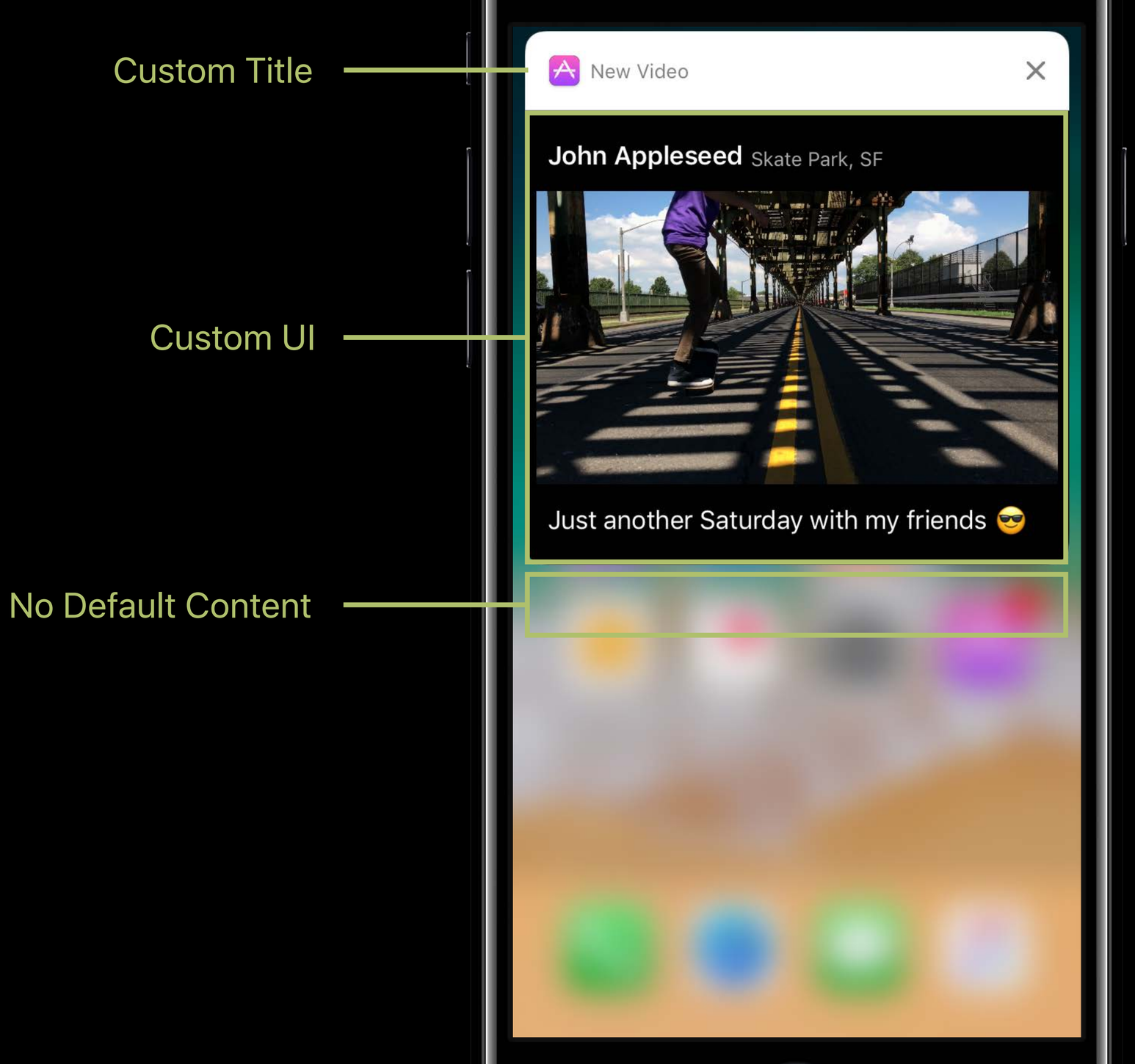
Custom Title

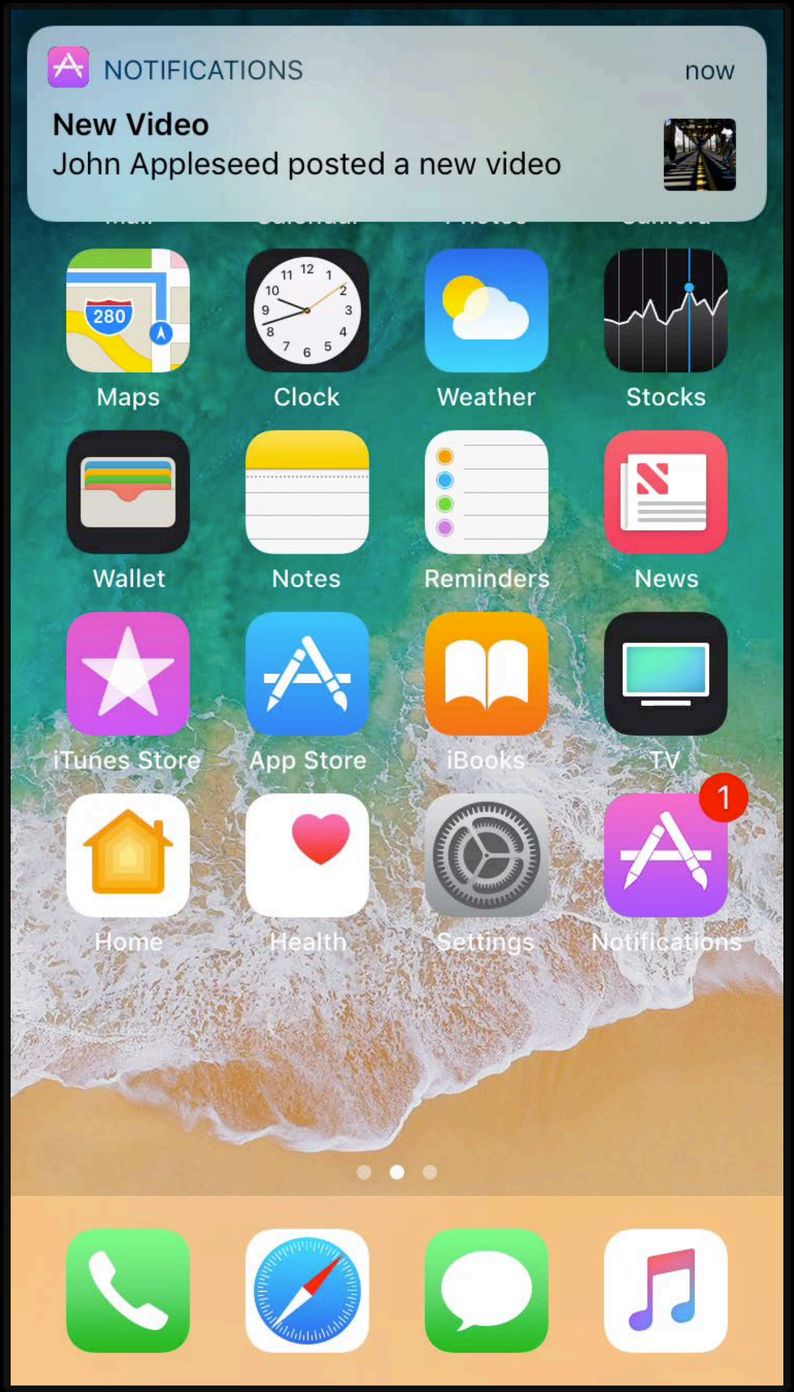


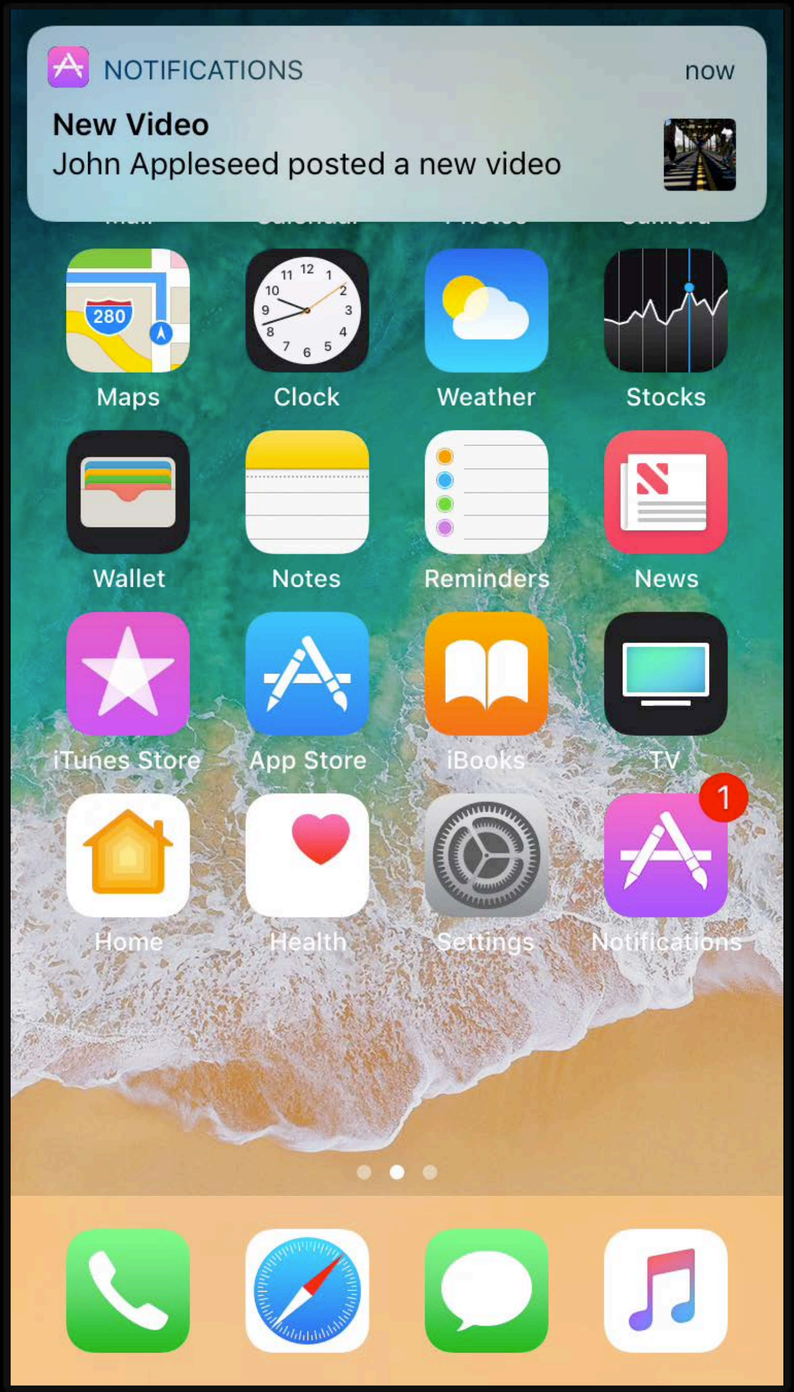
Custom Title

Custom UI




Custom Title
Custom UI
No Default Content
New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎


NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1


NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1

Customizing Rich Notifications

Content Extension size ratio

Customizing Rich Notifications

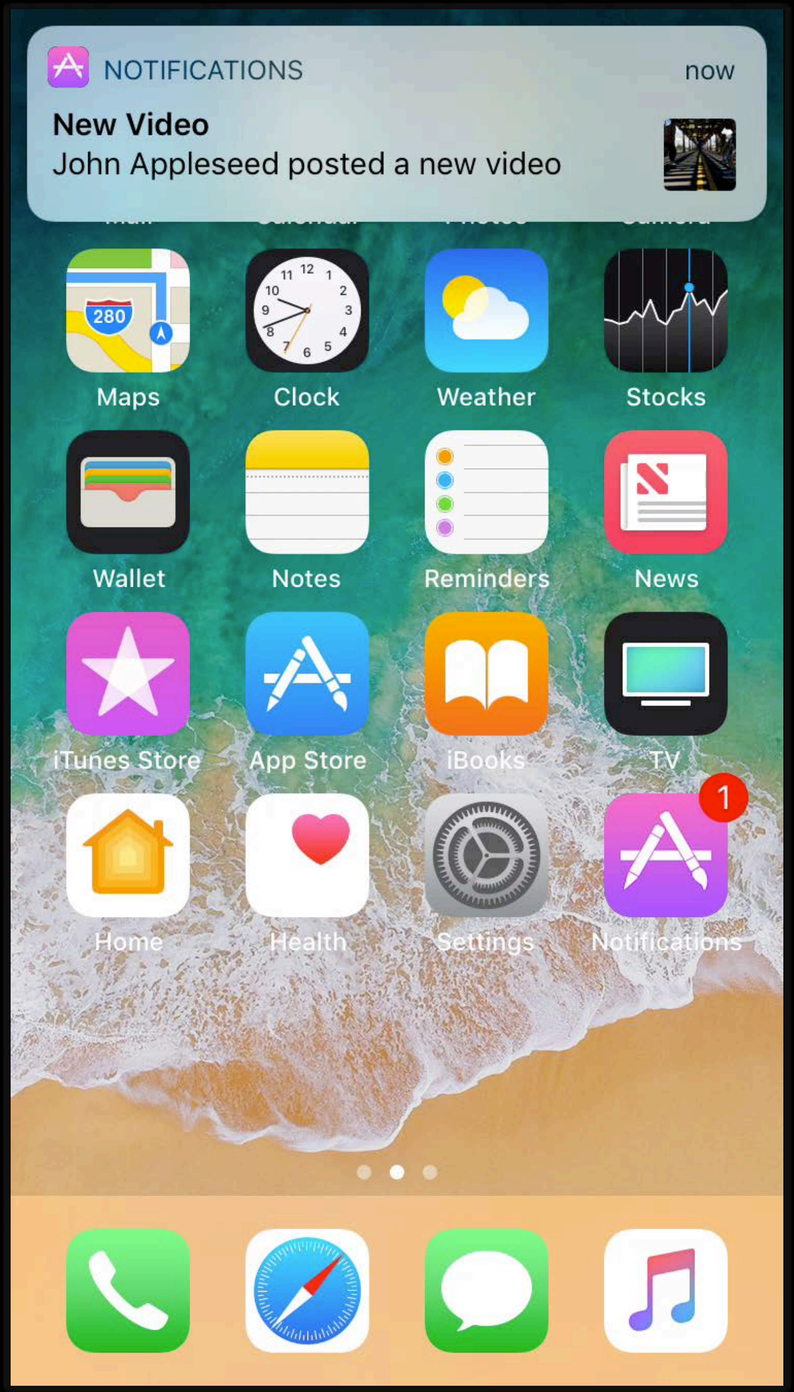
Content Extension size ratio

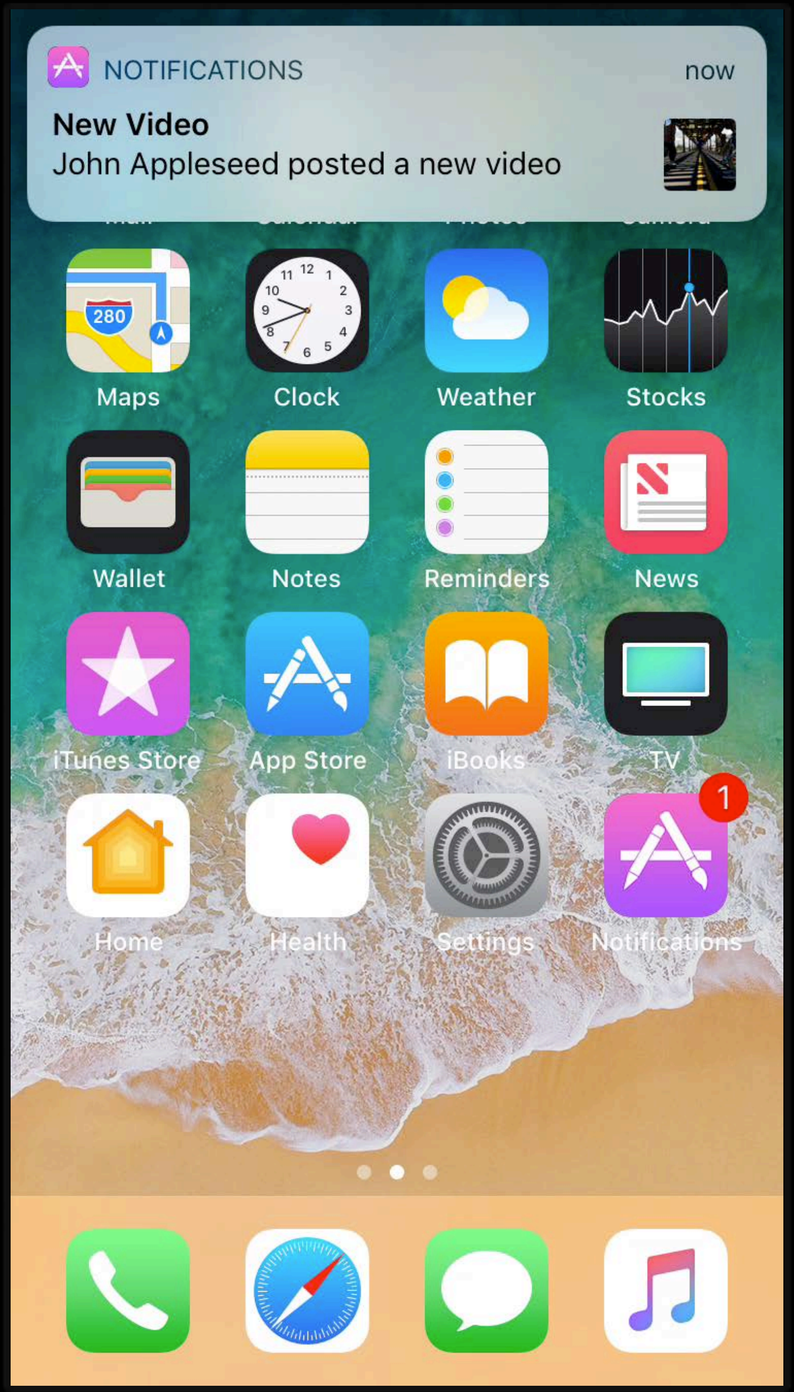
Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	NotificationMediaAttachmentExtension
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(4 items)
UNNotificationExtensionCategory	String	video-playback-category
UNNotificationExtensionDefaultContentHidden	Boolean	YES
UNNotificationExtensionOverridesDefaultTitle	Boolean	YES
UNNotificationExtensionInitialContentSizeRatio	Number	0.8
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension

Customizing Rich Notifications

Content Extension size ratio

Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	NotificationMediaAttachmentExtension
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(4 items)
UNNotificationExtensionCategory	String	video-playback-category
UNNotificationExtensionDefaultContentHidden	Boolean	YES
UNNotificationExtensionOverridesDefaultTitle	Boolean	YES
UNNotificationExtensionInitialContentSizeRatio	Number	0.8
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension


NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1


NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1

Customize Rich Notifications

Best practices

Customize Rich Notifications

Best practices

Customize Rich Notifications

Best practices

Custom UI elements

Customize Rich Notifications

Best practices

Custom UI elements

Display all relevant information

Customize Rich Notifications

Best practices

Custom UI elements

Display all relevant information

Reuse app view controllers

Customize Rich Notifications

Best practices

Custom UI elements

Display all relevant information

Reuse app view controllers

Correct sizing

Customize Rich Notifications

Best practices

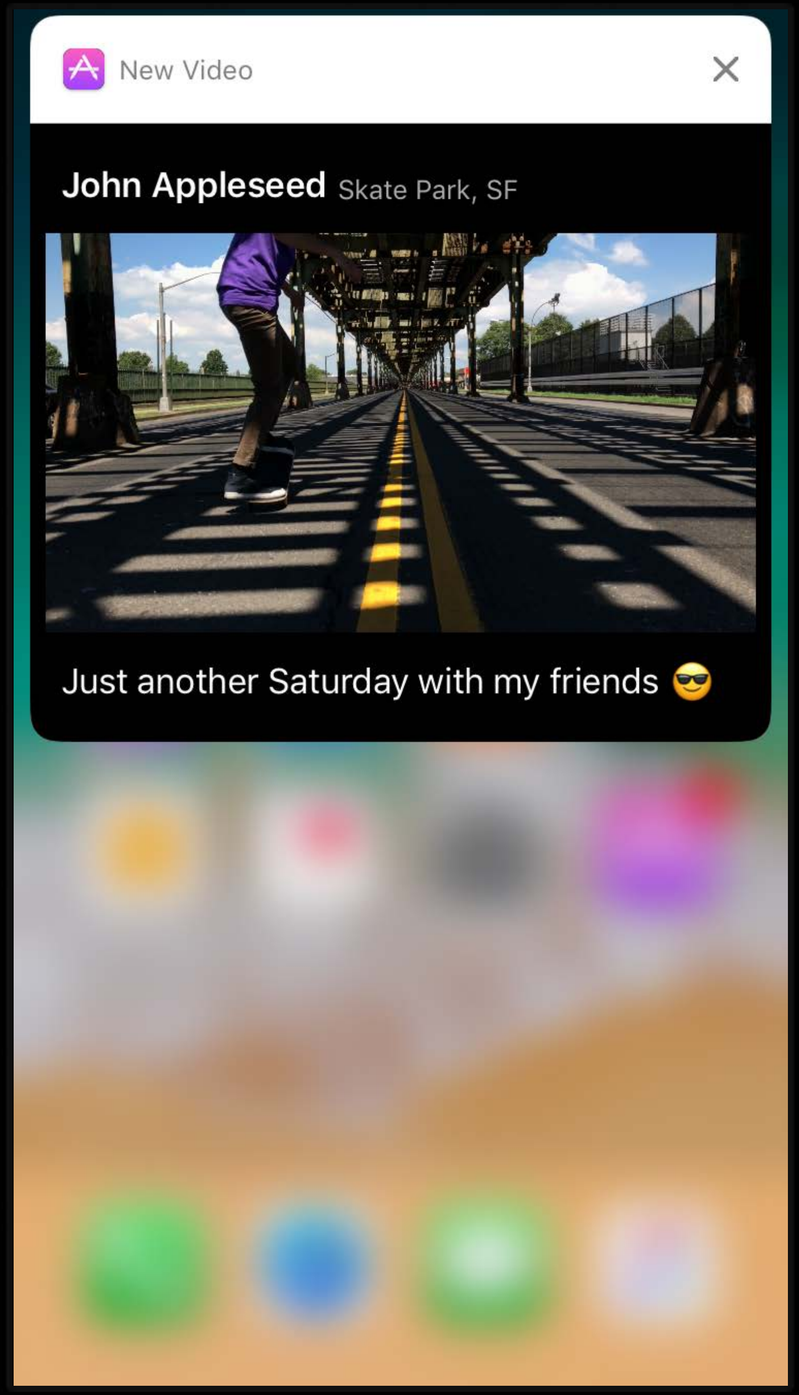
Custom UI elements

Display all relevant information

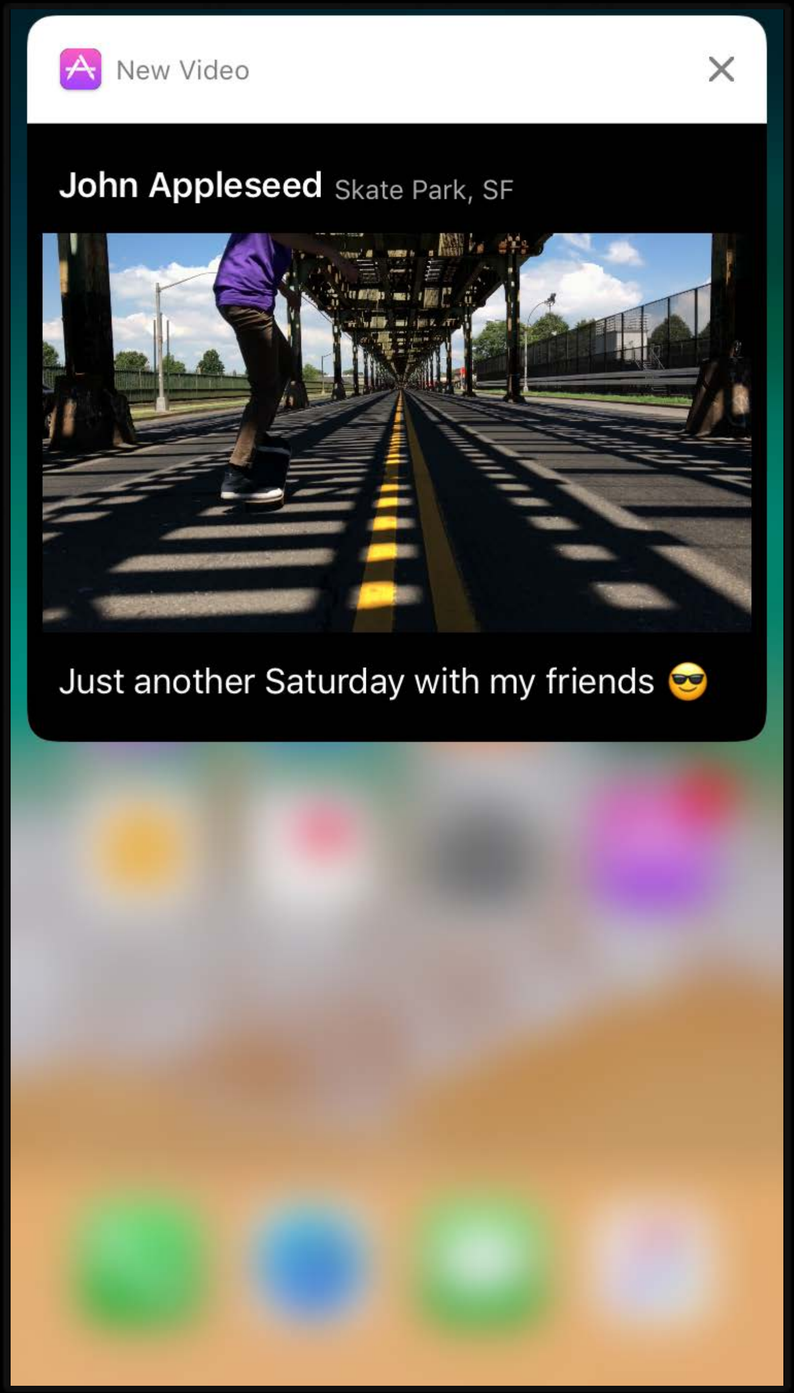
Reuse app view controllers

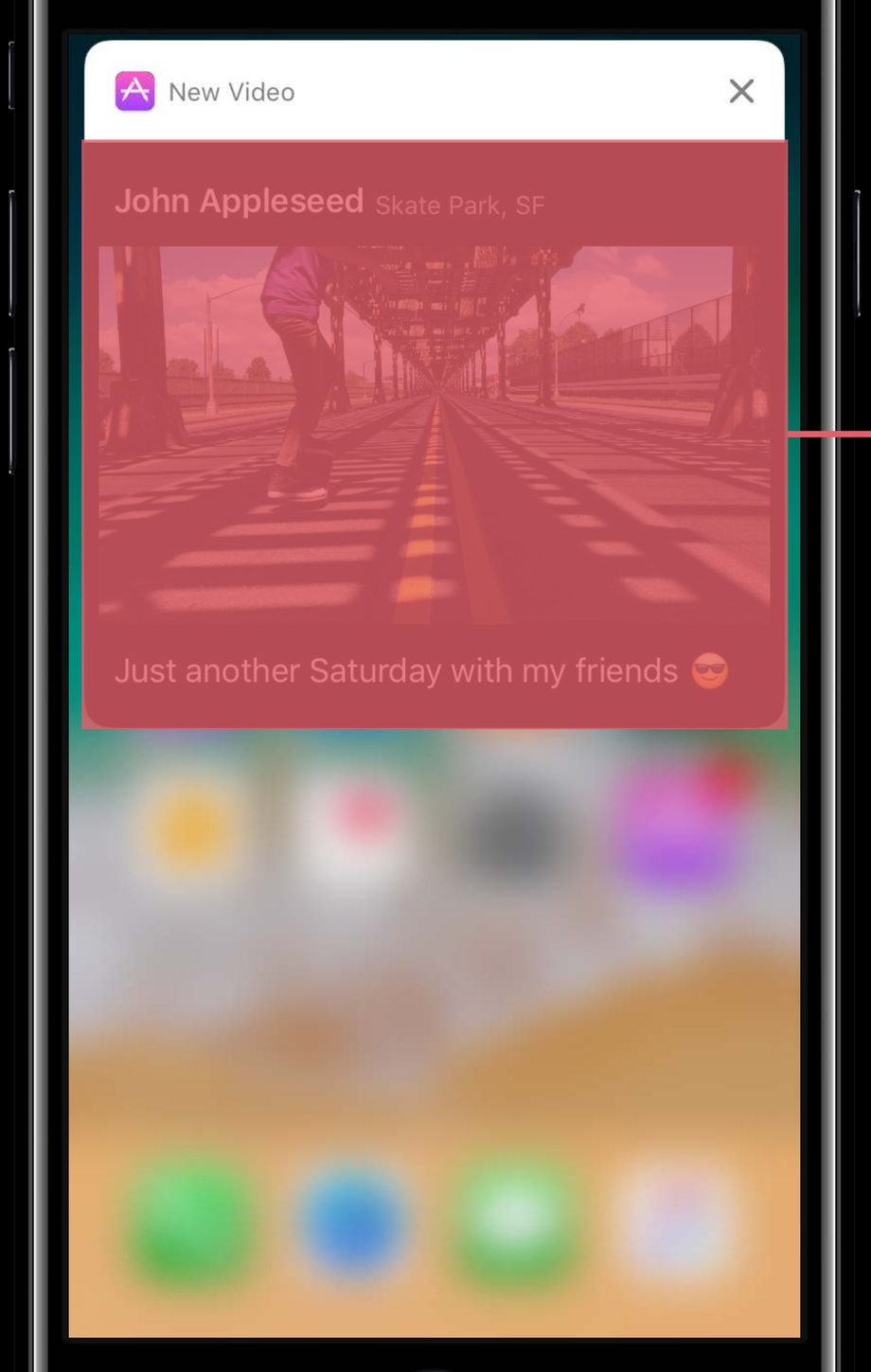
Correct sizing

Fast loading and layout


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎

User Input Customization


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎

No Touches

User Input Customization

User Input Customization

Media buttons

User Input Customization

Media buttons

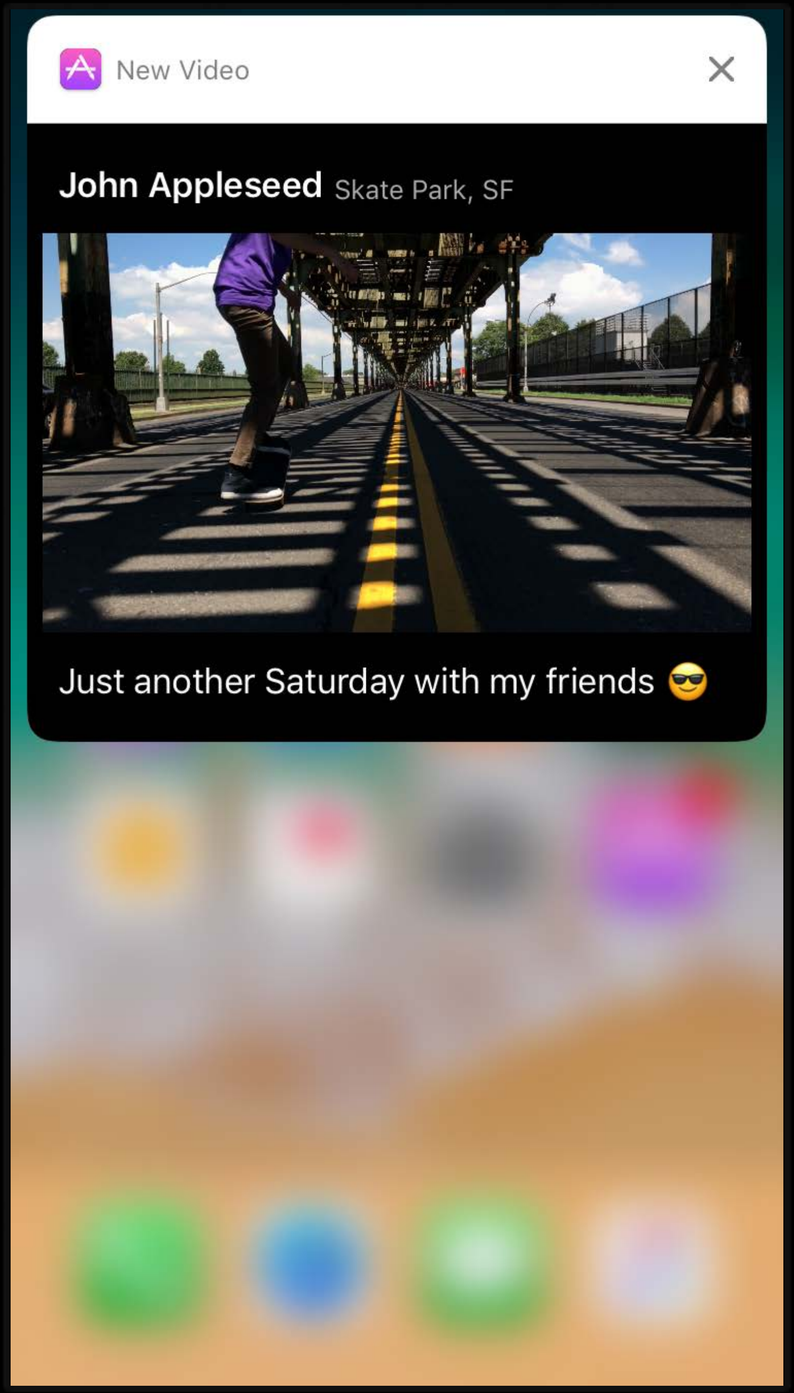
Actions

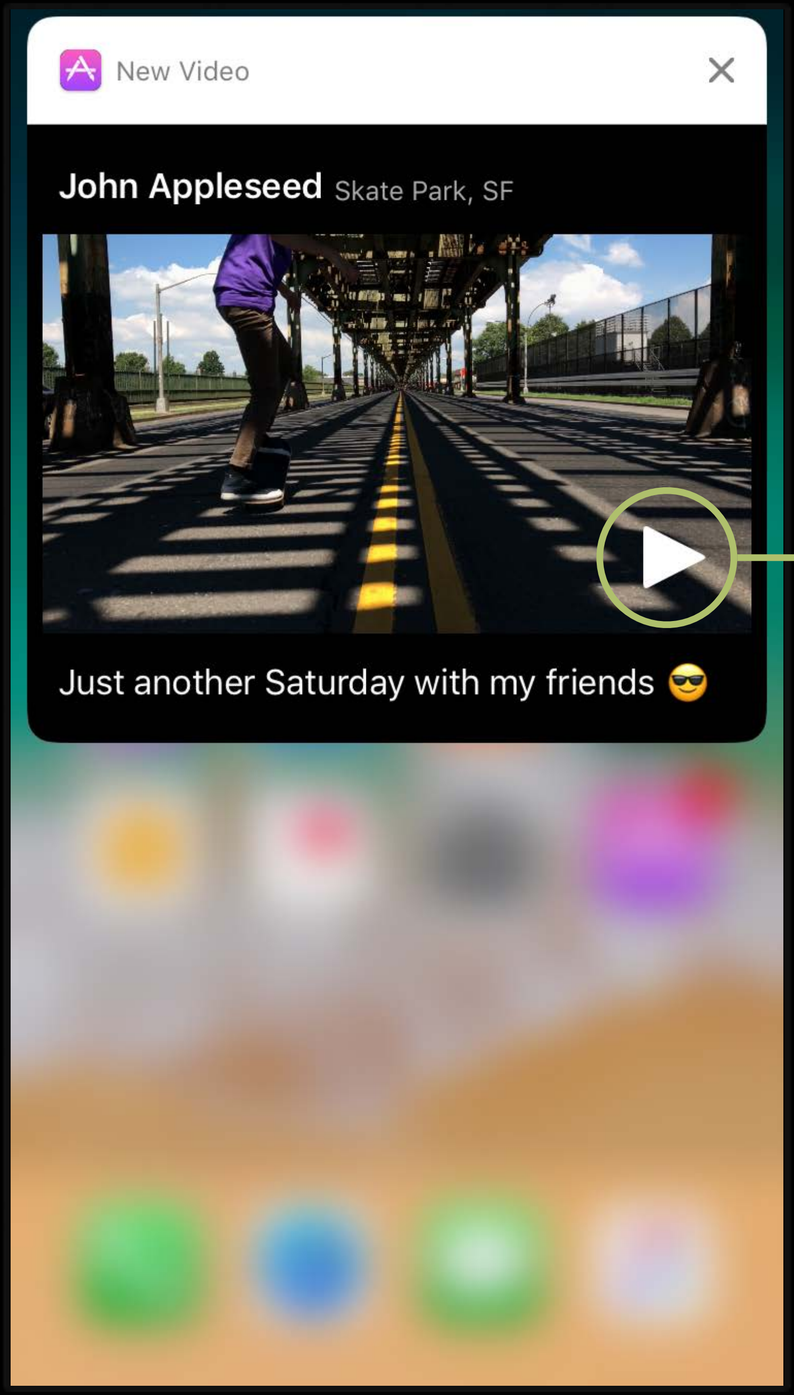
User Input Customization

Media buttons

Actions

Custom input views


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎

Play Button

```swift
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var mediaPlayPauseButtonFrame: CGRect {
        return CGRect(x: buttonX, y: buttonY, width: buttonWidth, height: buttonHeight)
    }

    var mediaPlayPauseButtonTintColor: UIColor {
        return .white
    }

    var mediaPlayPauseButtonType: UNNotificationContentExtensionMediaPlayPauseButtonType {
        return .default
    }
}
```

```swift
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var mediaPlayPauseButtonFrame: CGRect {
        return CGRect(x: buttonX, y: buttonY, width: buttonWidth, height: buttonHeight)
    }

    var mediaPlayPauseButtonTintColor: UIColor {
        return .white
    }

    var mediaPlayPauseButtonType: UNNotificationContentExtensionMediaPlayPauseButtonType {
        return .default
    }
}
```

```swift
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var mediaPlayPauseButtonFrame: CGRect {
        return CGRect(x: buttonX, y: buttonY, width: buttonWidth, height: buttonHeight)
    }

    var mediaPlayPauseButtonTintColor: UIColor {
        return .white
    }

    var mediaPlayPauseButtonType: UNNotificationContentExtensionMediaPlayPauseButtonType {
        return .default
    }
}
```

```swift
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var mediaPlayPauseButtonFrame: CGRect {
        return CGRect(x: buttonX, y: buttonY, width: buttonWidth, height: buttonHeight)
    }

    var mediaPlayPauseButtonTintColor: UIColor {
        return .white
    }

    var mediaPlayPauseButtonType: UNNotificationContentExtensionMediaPlayPauseButtonType {
        return .default
    }
}
```

```swift
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var mediaPlayPauseButtonFrame: CGRect {
        return CGRect(x: buttonX, y: buttonY, width: buttonWidth, height: buttonHeight)
    }

    var mediaPlayPauseButtonTintColor: UIColor {
        return .white
    }

    var mediaPlayPauseButtonType: UNNotificationContentExtensionMediaPlayPauseButtonType {
        return .default
    }
}
```

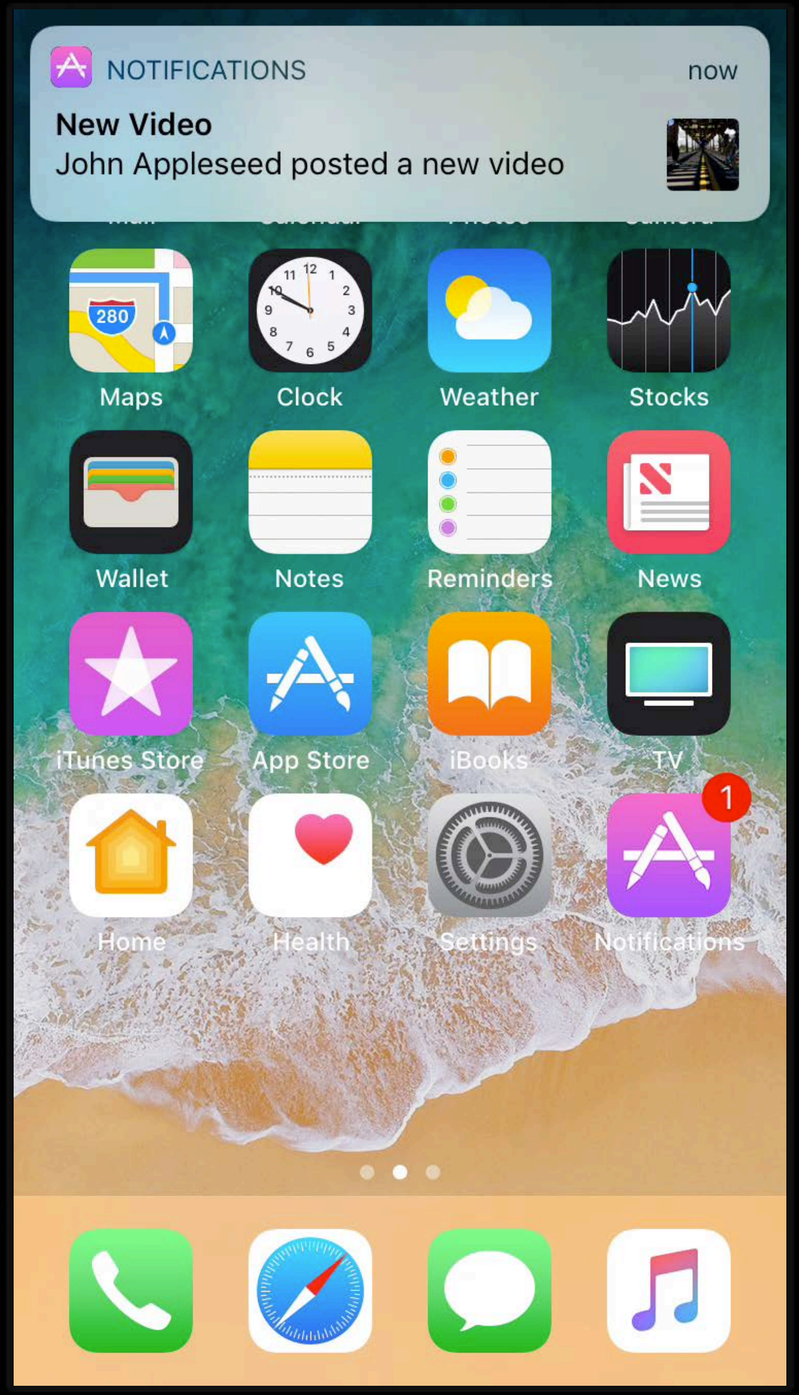
```swift
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var mediaPlayPauseButtonFrame: CGRect {
        return CGRect(x: buttonX, y: buttonY, width: buttonWidth, height: buttonHeight)
    }

    var mediaPlayPauseButtonTintColor: UIColor {
        return .white
    }

    var mediaPlayPauseButtonType: UNNotificationContentExtensionMediaPlayPauseButtonType {
        return .default
    }
}
```

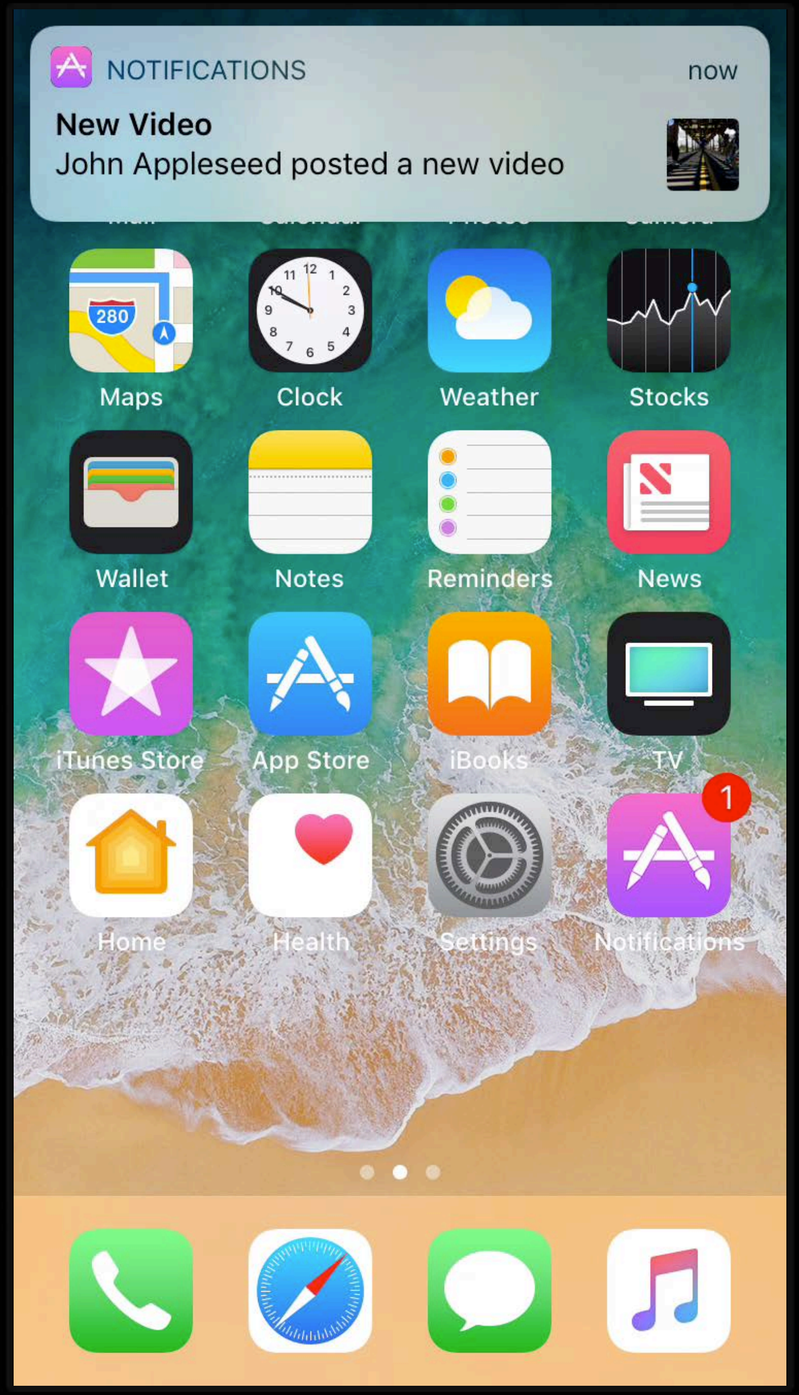
```
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func mediaPlay() {
        self.videoPlayer?.play()
    }

    func mediaPause() {
        self.videoPlayer?.pause()
    }

}
```

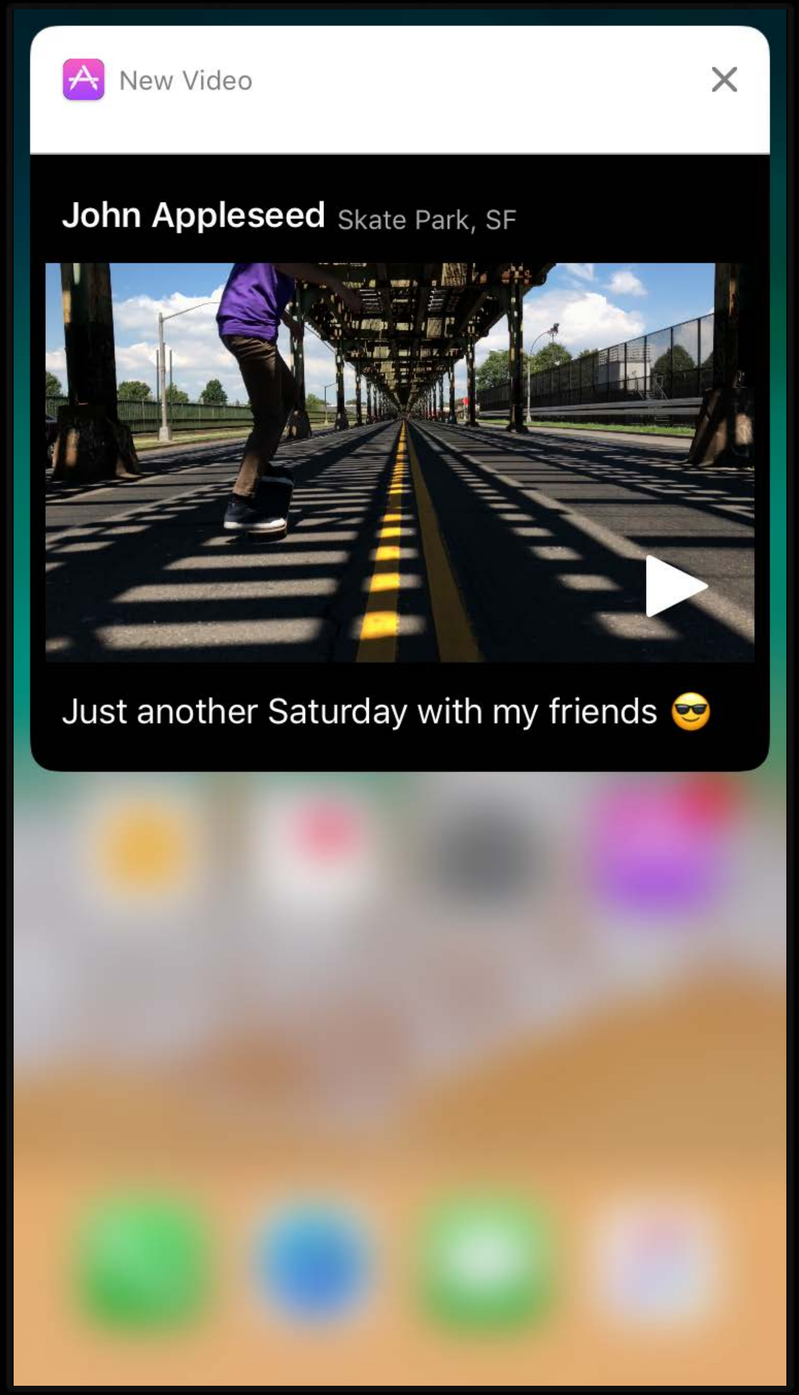
```swift
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func mediaPlay() {
        self.videoPlayer?.play()
    }

    func mediaPause() {
        self.videoPlayer?.pause()
    }

}
```

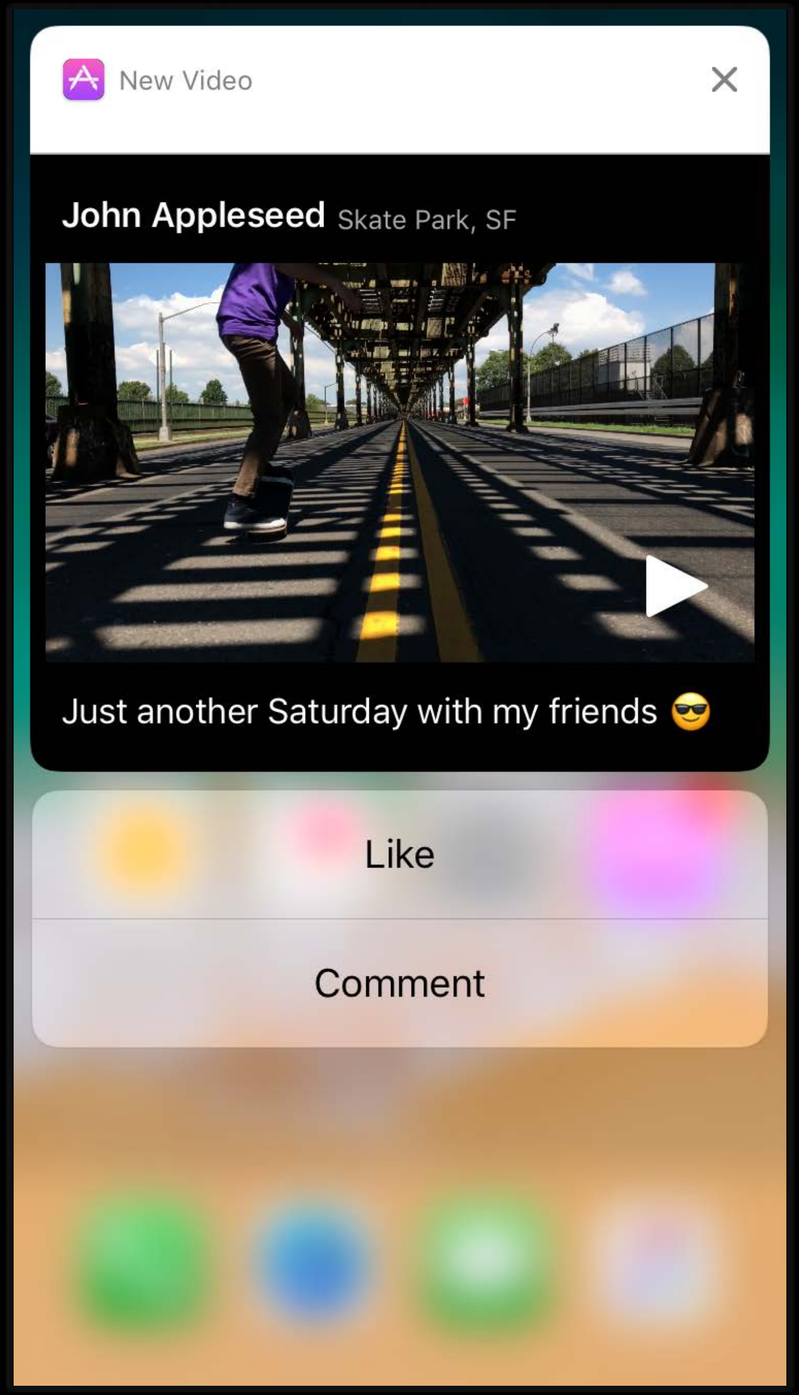
```swift
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func mediaPlay() {
        self.videoPlayer?.play()
    }

    func mediaPause() {
        self.videoPlayer?.pause()
    }

}
```

```
// User Input Customization - Media Controls
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func mediaPlay() {
        self.videoPlayer?.play()
    }

    func mediaPause() {
        self.videoPlayer?.pause()
    }

}
```


NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1

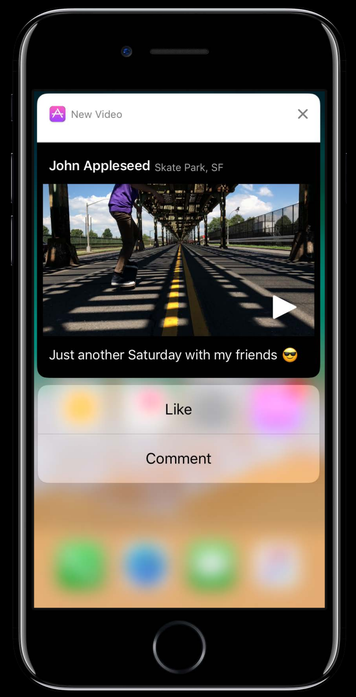

NOTIFICATIONS
now
New Video
John Appleseed posted a new video
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Notifications
1


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
Like
Comment

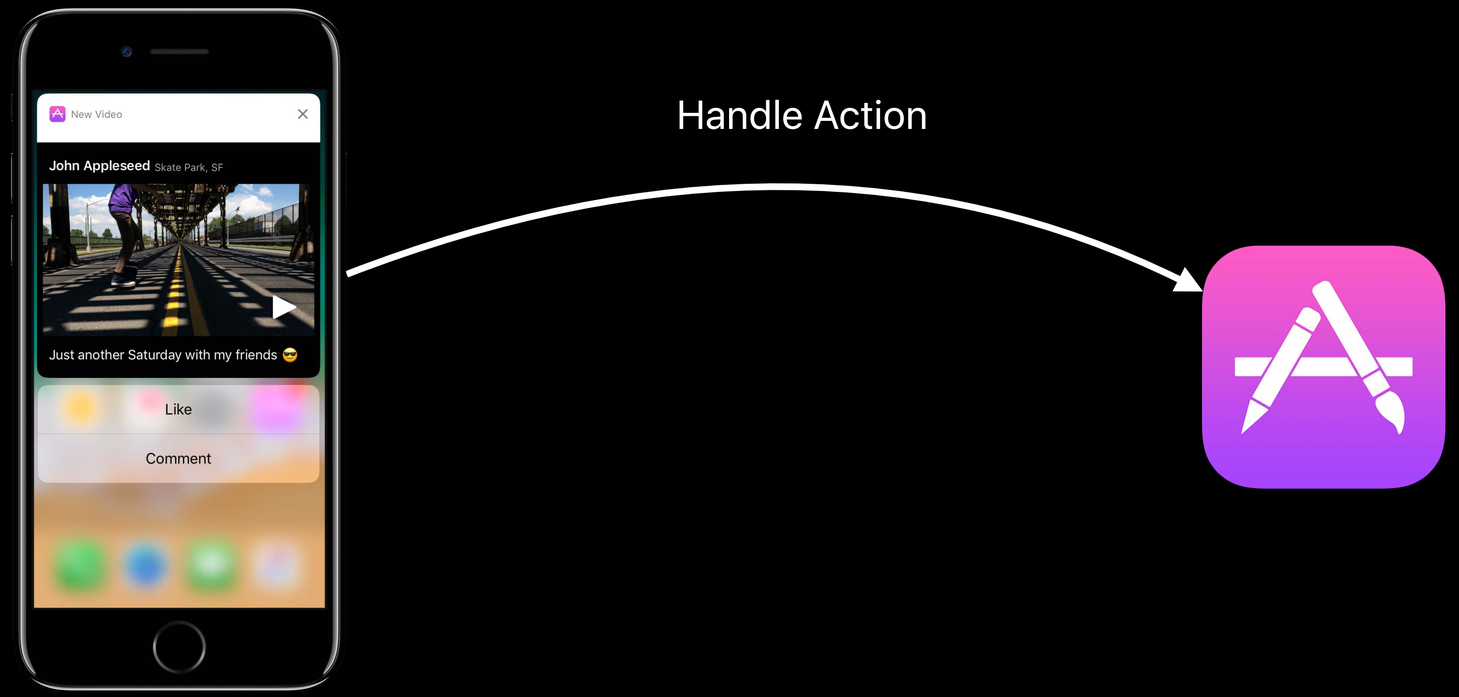
User Input Customization

Action handling



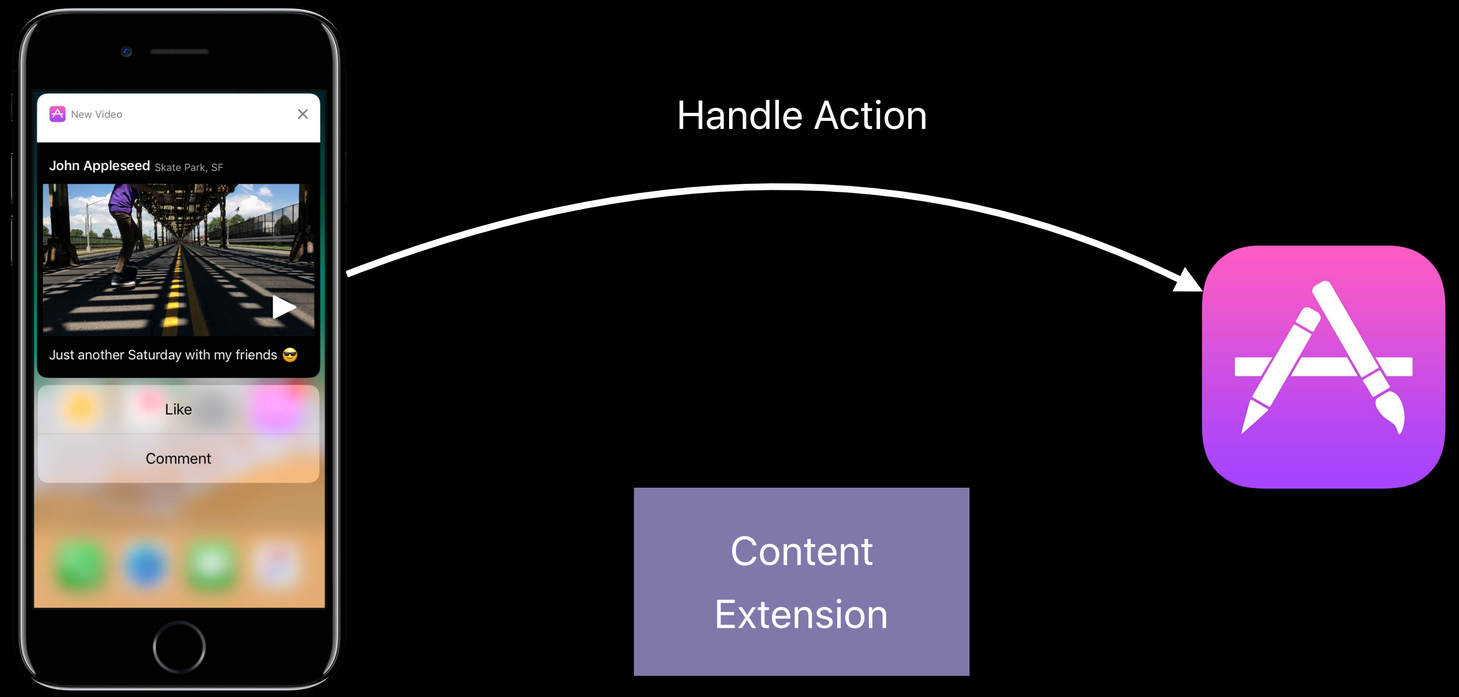
User Input Customization

Action handling



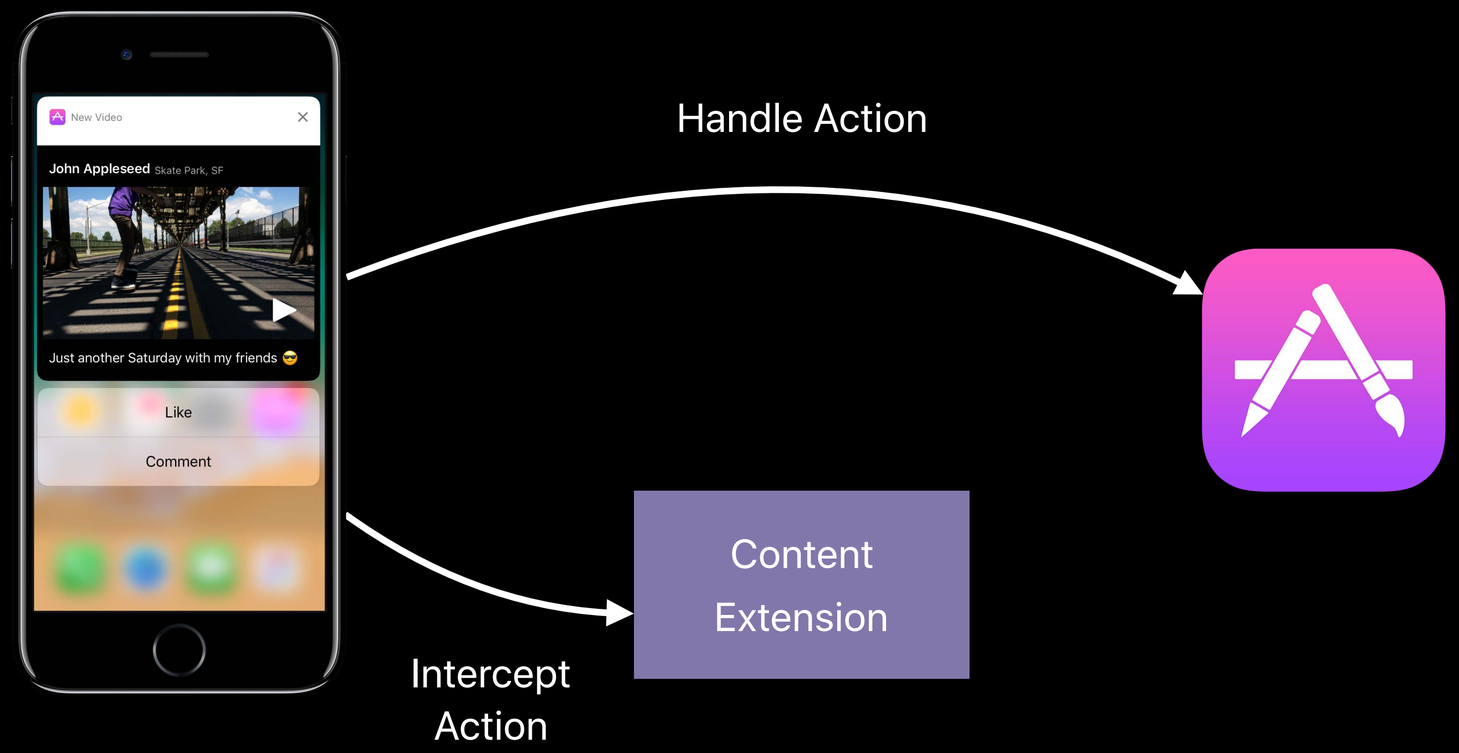
User Input Customization

Action handling



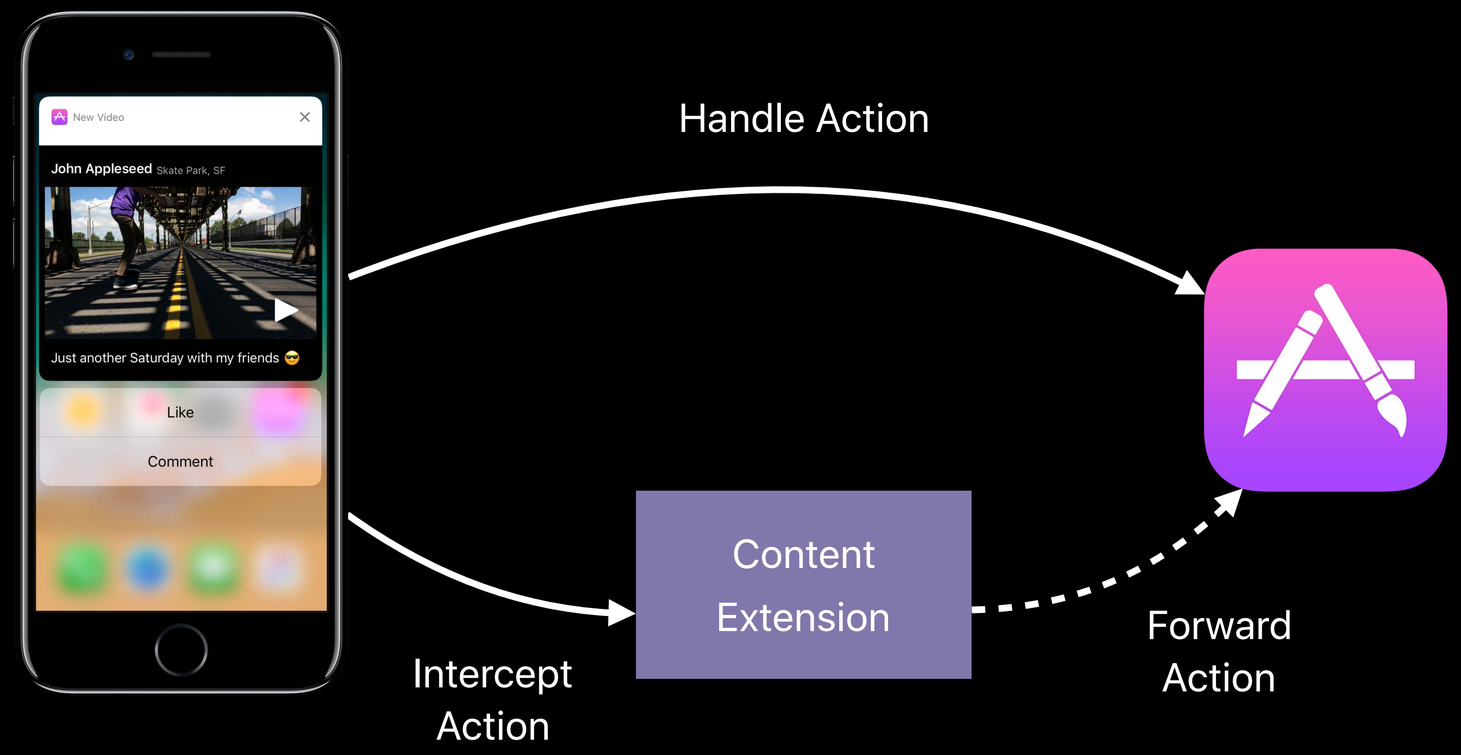
User Input Customization

Action handling



User Input Customization

Action handling



```
// User Input Customization - Action Handling

let likeAction = UNNotificationAction(identifier: "like-action",
                                      title: "Like")

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
                                                  title: "Comment",
                                                  textInputButtonTitle: "Post",
                                                  textInputPlaceholder: "Comment")

let category = UNNotificationCategory(identifier: "video-playback-category",
                                      actions: [ likeAction, commentAction ],
                                      intentIdentifiers: [])
```

```
// User Input Customization - Action Handling

let likeAction = UNNotificationAction(identifier: "like-action",
                                      title: "Like")

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
                                                  title: "Comment",
                                                  textInputButtonTitle: "Post",
                                                  textInputPlaceholder: "Comment")

let category = UNNotificationCategory(identifier: "video-playback-category",
                                      actions: [ likeAction, commentAction ],
                                      intentIdentifiers: [])
```

```
// User Input Customization - Action Handling

let likeAction = UNNotificationAction(identifier: "like-action",
                                      title: "Like")

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
                                                  title: "Comment",
                                                  textInputButtonTitle: "Post",
                                                  textInputPlaceholder: "Comment")

let category = UNNotificationCategory(identifier: "video-playback-category",
                                      actions: [ likeAction, commentAction ],
                                      intentIdentifiers: [])
```

```
// User Input Customization - Action Handling

let likeAction = UNNotificationAction(identifier: "like-action",
                                      title: "Like")

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
                                                  title: "Comment",
                                                  textInputButtonTitle: "Post",
                                                  textInputPlaceholder: "Comment")

let category = UNNotificationCategory(identifier: "video-playback-category",
                                      actions: [ likeAction, commentAction ],
                                      intentIdentifiers: [])
```

```
// User Input Customization - Action Handling

let likeAction = UNNotificationAction(identifier: "like-action",
                                      title: "Like")

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
                                                  title: "Comment",
                                                  textInputButtonTitle: "Post",
                                                  textInputPlaceholder: "Comment")

let category = UNNotificationCategory(identifier: "video-playback-category",
                                      actions: [ likeAction, commentAction ],
                                      intentIdentifiers: [])
```

```swift
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "like-action" {
                self.likeLabel?.text = "👍"
        }

        completionHandler(.doNotDismiss)
    }

}
```

```
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "like-action" {
                self.likeLabel?.text = "👍"
        }

        completionHandler(.doNotDismiss)
    }

}
```

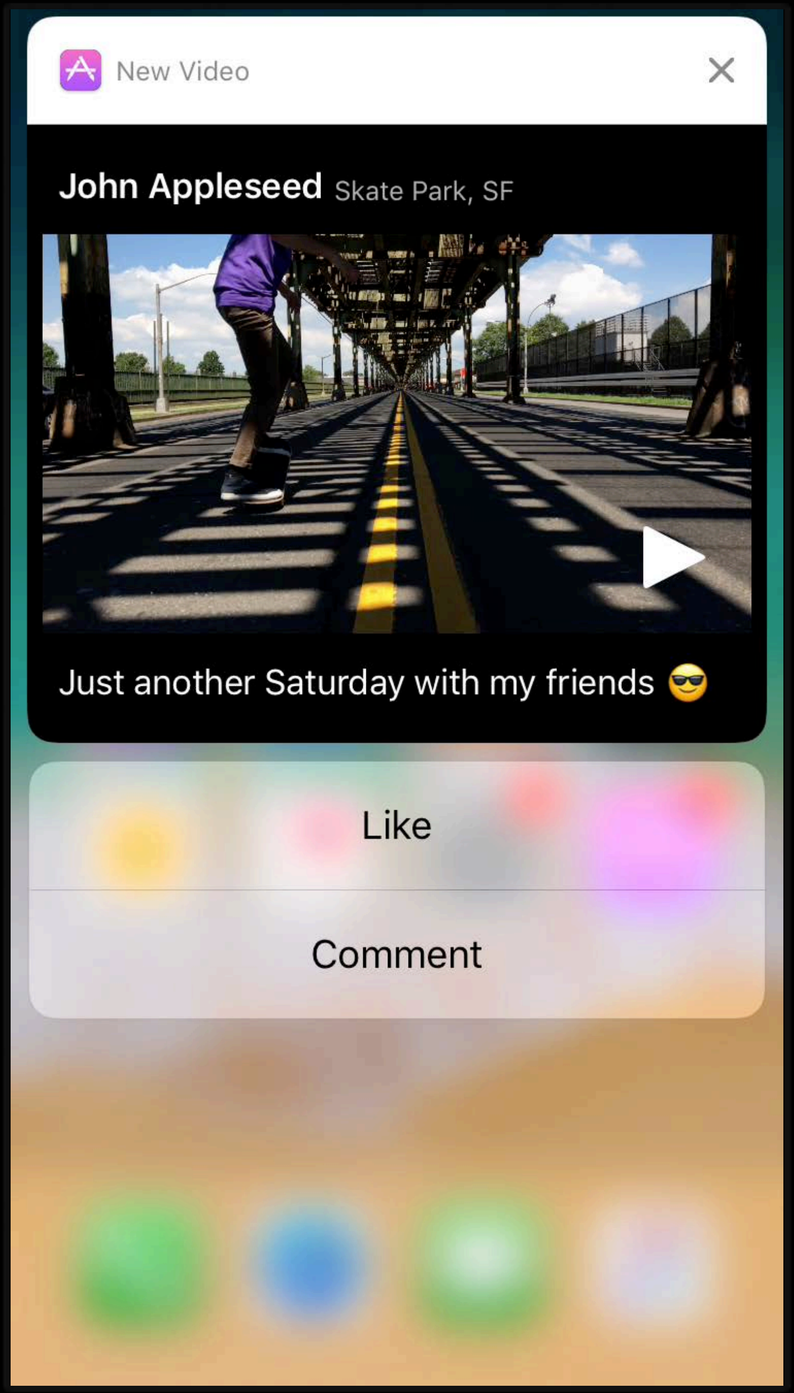
```swift
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "like-action" {
                self.likeLabel?.text = "👍"
        }

        completionHandler(.doNotDismiss)
    }

}
```

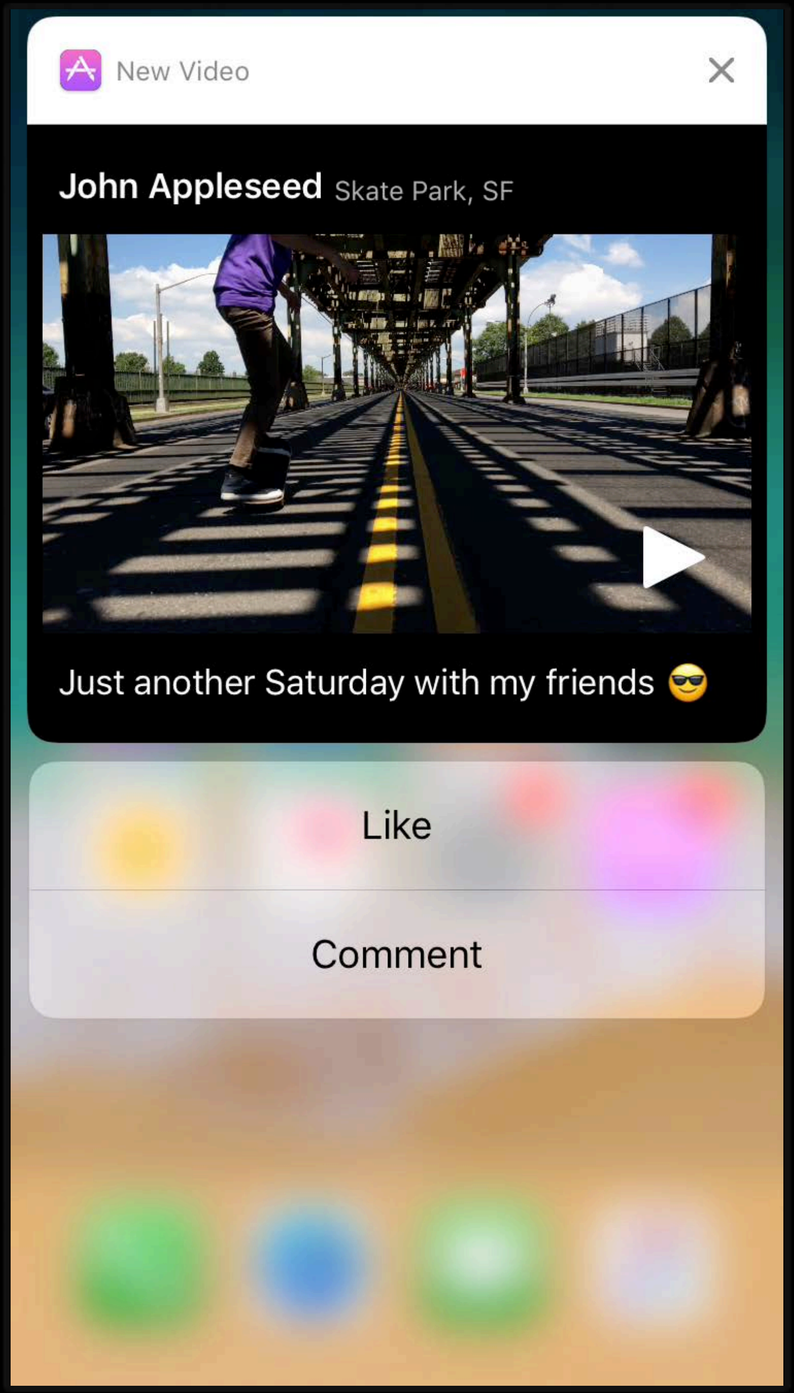
```
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "like-action" {
                self.likeLabel?.text = "👍"
        }

        completionHandler(.doNotDismiss)
    }

}
```

```
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "like-action" {
                self.likeLabel?.text = "👍"
        }

        completionHandler(.doNotDismiss)
    }

}
```

```
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "like-action" {
                self.likeLabel?.text = "👍"
        }

        completionHandler(.doNotDismiss)
    }

}
```

```swift
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "like-action" {
                self.likeLabel?.text = "👍"
        }

        completionHandler(.doNotDismiss)
    }

}
```


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
Like
Comment


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
Like
Comment

```
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

   var commentLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "comment-action" {
            if let textResponse = response as? UNTextInputNotificationResponse {
                self.commentLabel?.text = textResponse.userText
            }
        }
        completionHandler(.doNotDismiss)
    }
}
```

```
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

   var commentLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "comment-action" {
            if let textResponse = response as? UNTextInputNotificationResponse {
                self.commentLabel?.text = textResponse.userText
            }
        }
        completionHandler(.doNotDismiss)
    }
}
```

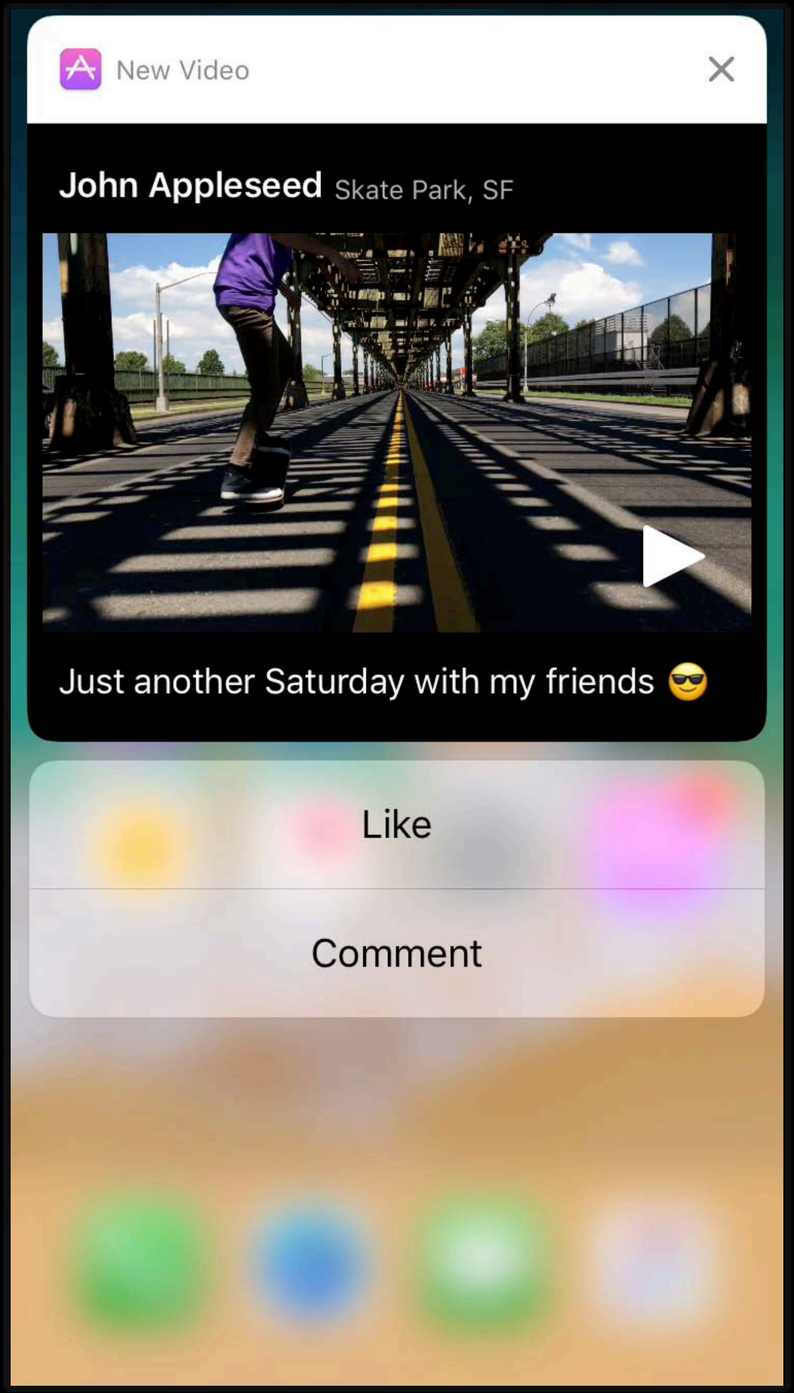
```swift
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

   var commentLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "comment-action" {
            if let textResponse = response as? UNTextInputNotificationResponse {
                self.commentLabel?.text = textResponse.userText
            }
        }
        completionHandler(.doNotDismiss)
    }
}
```

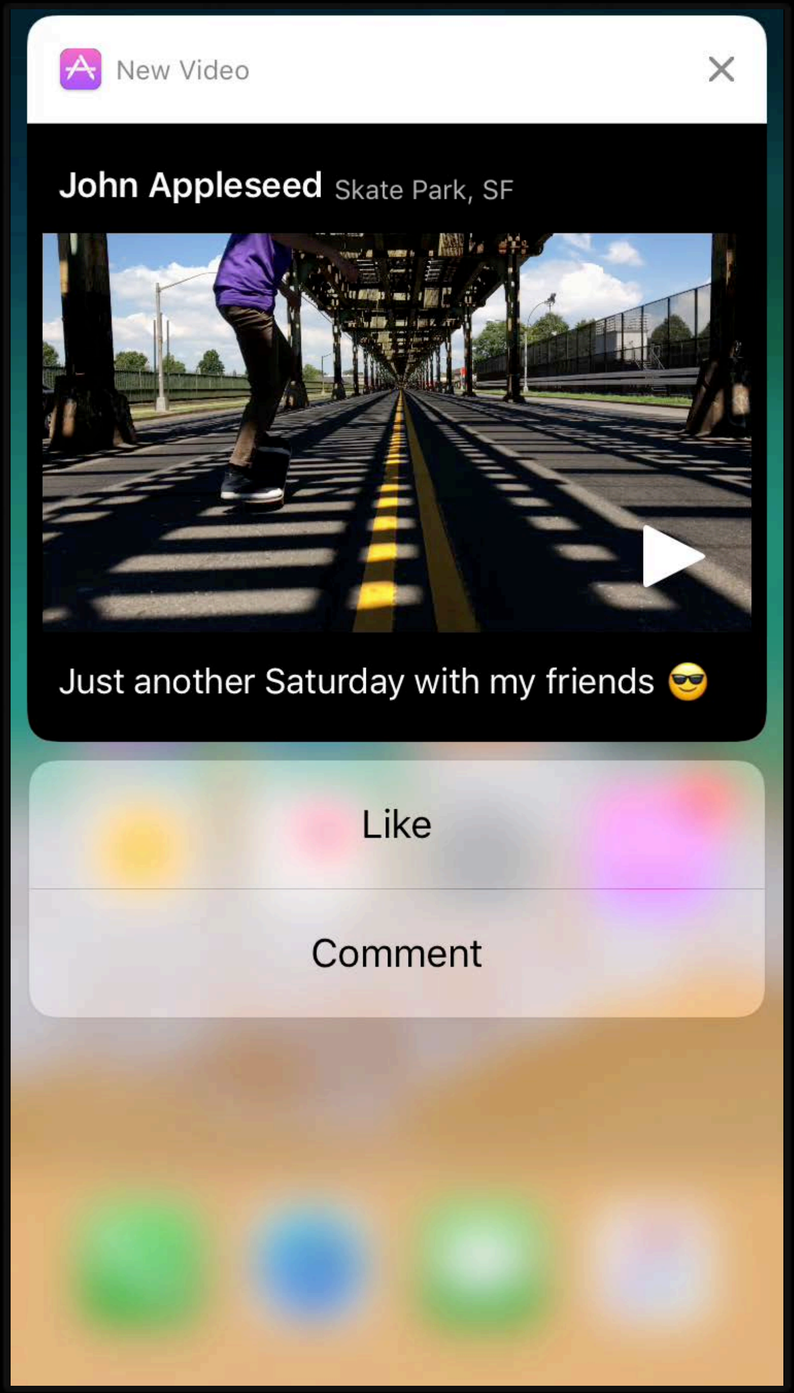
```
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

   var commentLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "comment-action" {
            if let textResponse = response as? UNTextInputNotificationResponse {
                self.commentLabel?.text = textResponse.userText
            }
        }
        completionHandler(.doNotDismiss)
    }
}
```

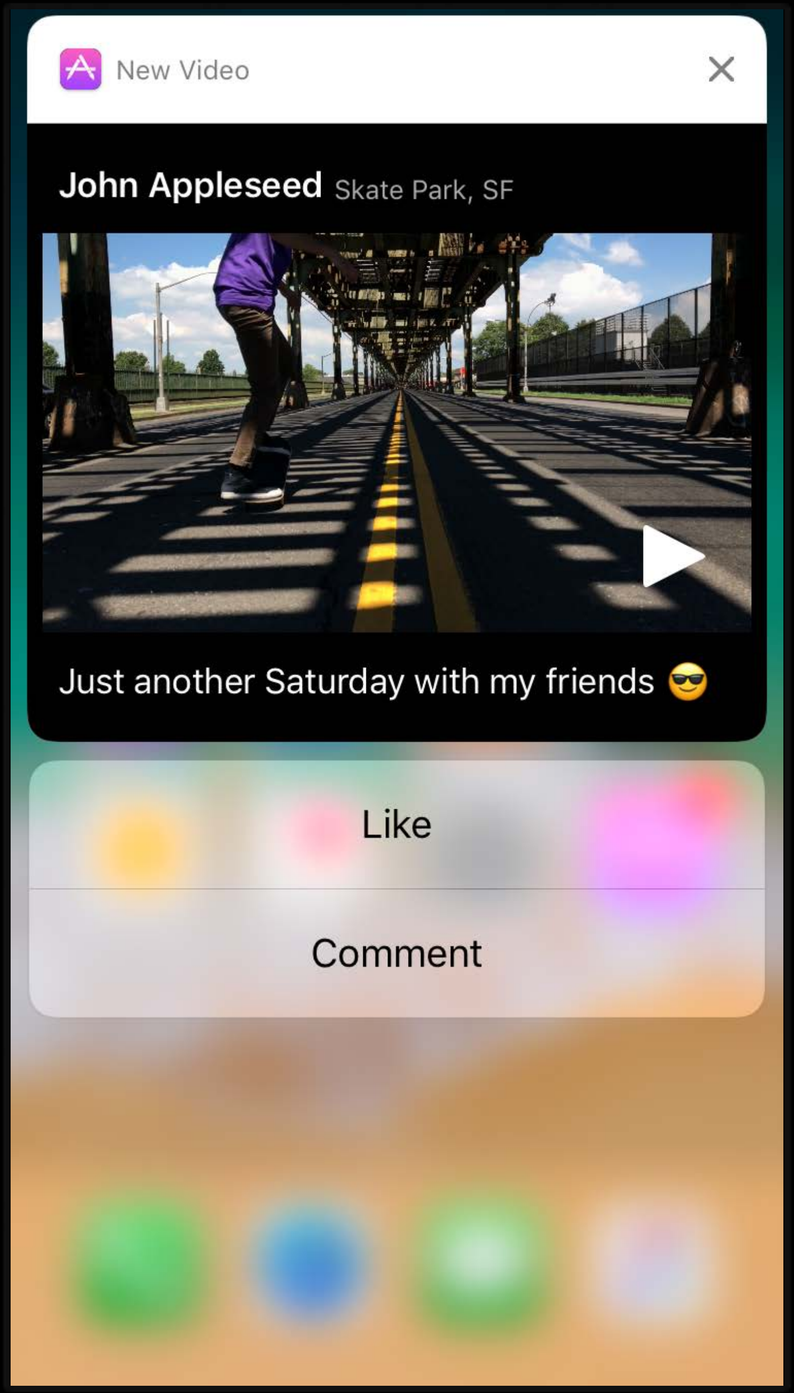
```swift
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var commentLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
        completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "comment-action" {
            if let textResponse = response as? UNTextInputNotificationResponse {
                self.commentLabel?.text = textResponse.userText
            }
        }
        completionHandler(.doNotDismiss)
    }
}
```

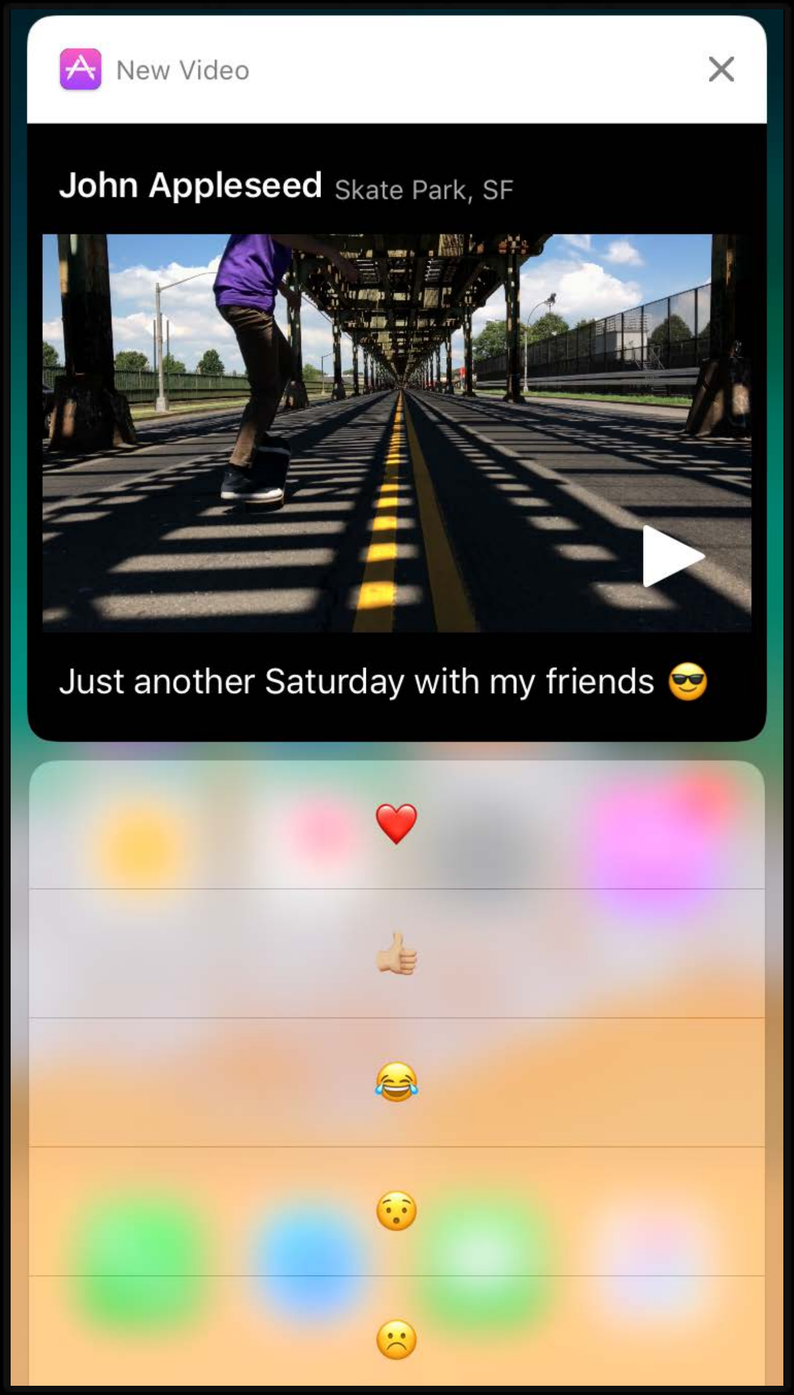
```
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

   var commentLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "comment-action" {
            if let textResponse = response as? UNTextInputNotificationResponse {
                self.commentLabel?.text = textResponse.userText
            }
        }
        completionHandler(.doNotDismiss)
    }
}
```

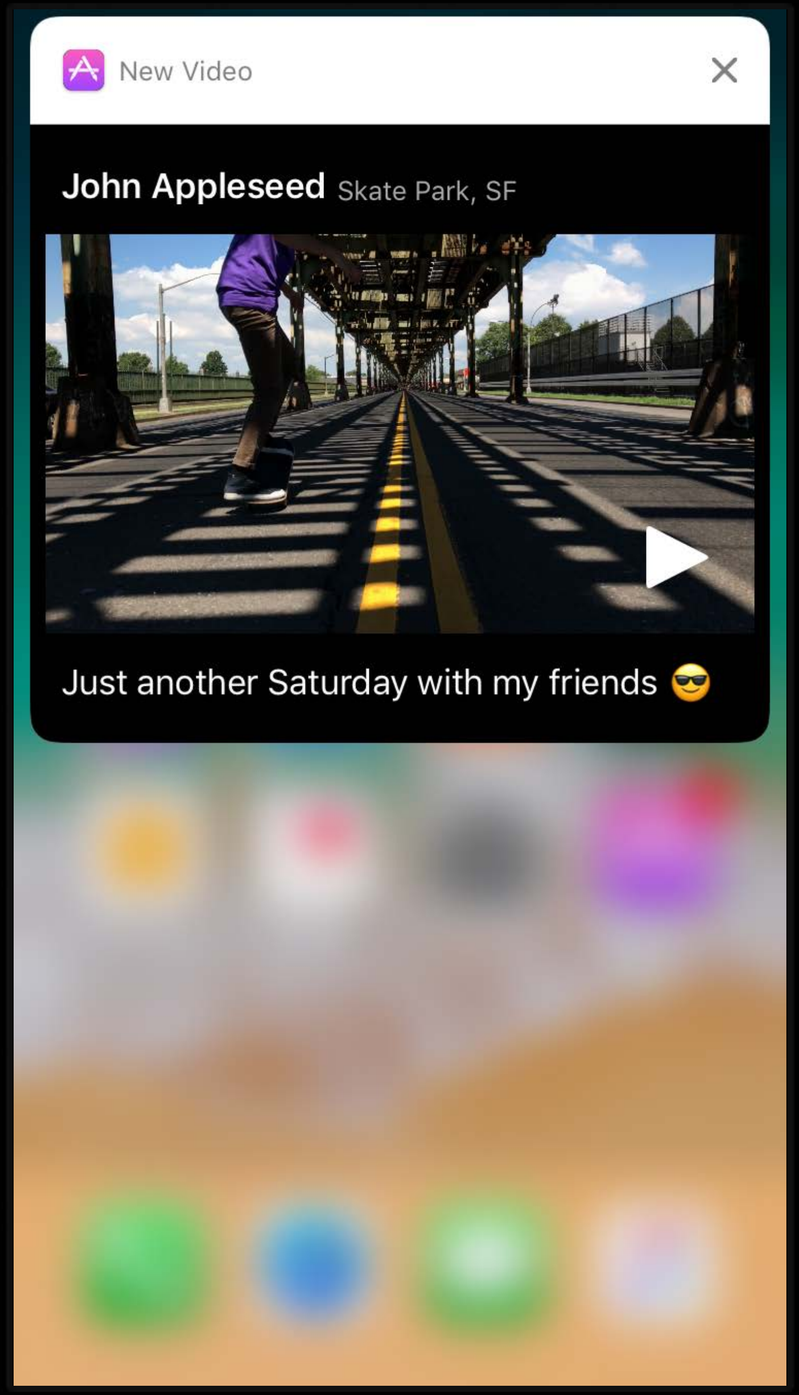
```
// User Input Customization - Action Handling

class NotificationViewController: UIViewController, UNNotificationContentExtension {

   var commentLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse,
       completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        server.postEventResponse(response)
        if response.actionIdentifier == "comment-action" {
            if let textResponse = response as? UNTextInputNotificationResponse {
                self.commentLabel?.text = textResponse.userText
            }
        }
        completionHandler(.doNotDismiss)
    }
}
```

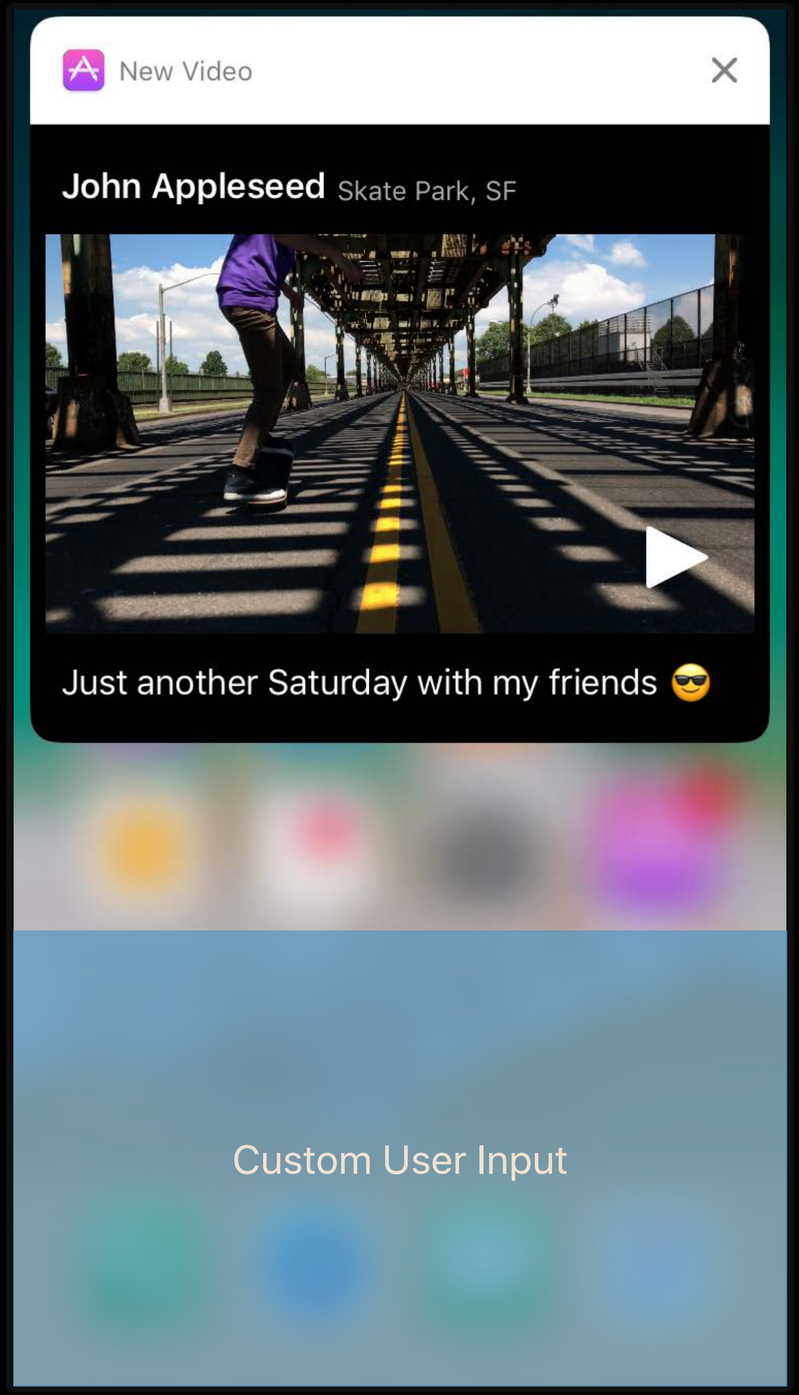

New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
Like
Comment

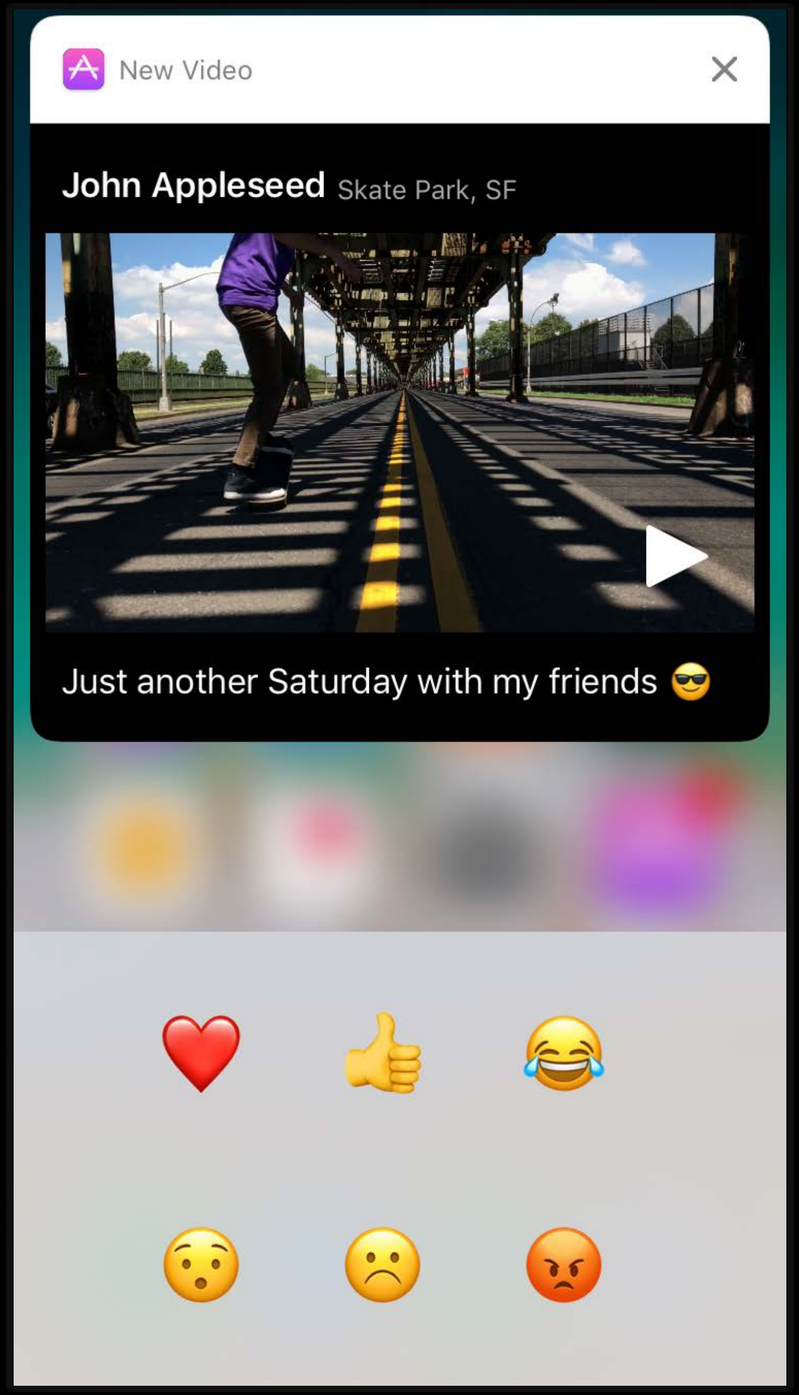

New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
Like
Comment


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
Like
Comment


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
❤️
👍🏼
😂
😯
☹️


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
Custom User Input


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎

UIResponder

```
// User Input Customization - Custom Input View

let reactionsAction = UNNotificationAction(identifier: "reactions-action",
                                           title: "Reaction")

let category = UNNotificationCategory(identifier: "video-playback-category",
                                      actions: [ reactionsAction ],
                                      intentIdentifiers: [])
```

```
// User Input Customization – Custom Input View

let reactionsAction = UNNotificationAction(identifier: "reactions-action",
                                           title: "Reaction")

let category = UNNotificationCategory(identifier: "video-playback-category",
                                      actions: [ reactionsAction ],
                                      intentIdentifiers: [])
```

```
// User Input Customization – Custom Input View

let reactionsAction = UNNotificationAction(identifier: "reactions-action",
                                           title: "Reaction")

let category = UNNotificationCategory(identifier: "video-playback-category",
                                      actions: [ reactionsAction ],
                                      intentIdentifiers: [])
```

```
// User Input Customization - Custom Input View

let reactionsAction = UNNotificationAction(identifier: "reactions-action",
                                           title: "Reaction")

let category = UNNotificationCategory(identifier: "video-playback-category",
                                      actions: [ reactionsAction ],
                                      intentIdentifiers: [])
```

```
// User Input Customization - Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var customInputView: UIView

    override public var canBecomeFirstResponder : Bool {
        return true
    }

    override var inputView: UIView {
        return self.customInputView
    }

}
```

```
// User Input Customization - Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var customInputView: UIView

    override public var canBecomeFirstResponder : Bool {
        return true
    }

    override var inputView: UIView {
        return self.customInputView
    }

}
```

```
// User Input Customization – Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var customInputView: UIView

    override public var canBecomeFirstResponder : Bool {
        return true
    }

    override var inputView: UIView {
        return self.customInputView
    }

}
```

```
// User Input Customization - Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    var customInputView: UIView

    override public var canBecomeFirstResponder : Bool {
        return true
    }

    override var inputView: UIView {
        return self.customInputView
    }

}
```

```
// User Input Customization — Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler
        completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "reactions-action" {
            self.becomeFirstResponder()
        }
    }

    func handleReactionButtonPress(sender: UIButton) {
        self.reactionLabel?.text = sender.currentTitle
        self.resignFirstResponder()
    }
}
```

```swift
// User Input Customization — Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler
        completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "reactions-action" {
            self.becomeFirstResponder()
        }
    }

    func handleReactionButtonPress(sender: UIButton) {
        self.reactionLabel?.text = sender.currentTitle
        self.resignFirstResponder()
    }
}
```

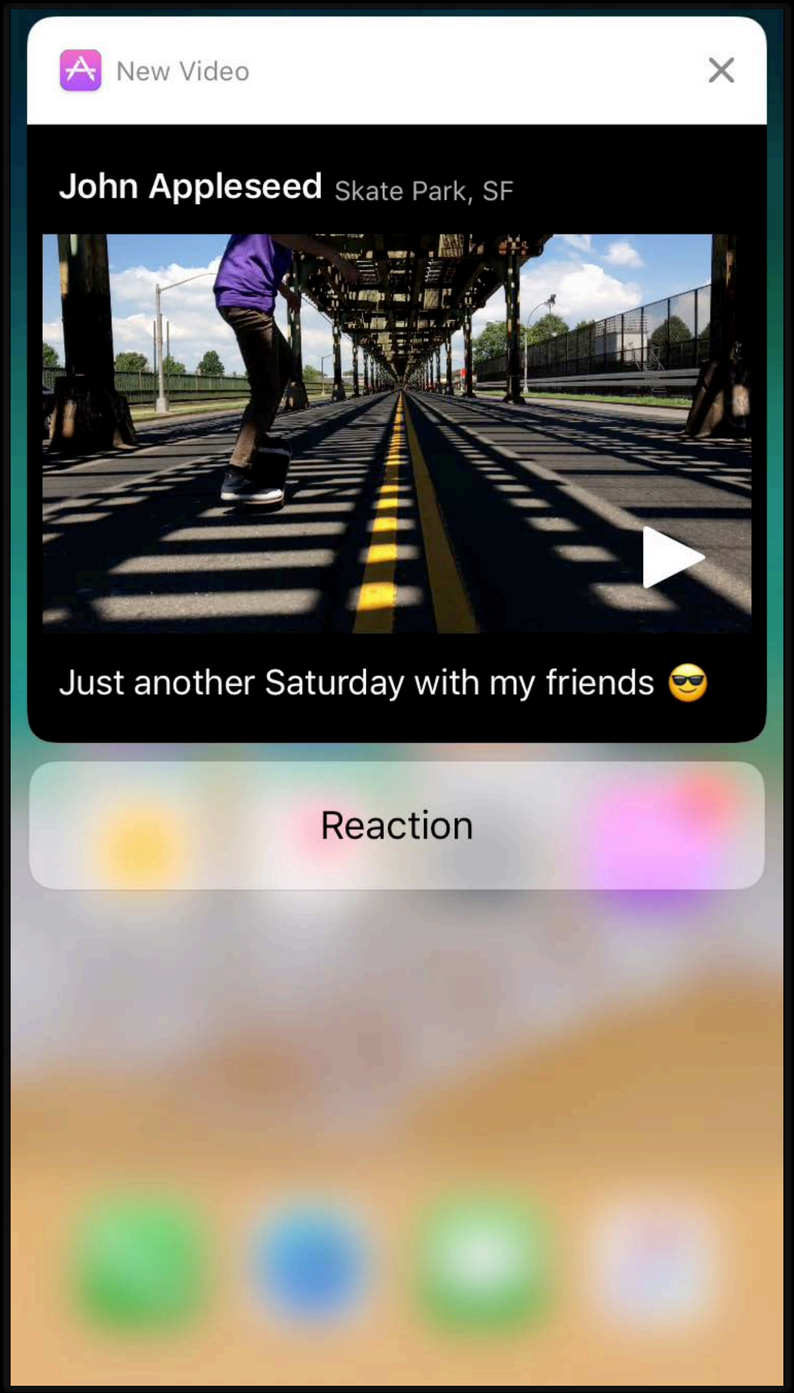
```
// User Input Customization — Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler
        completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "reactions-action" {
            self.becomeFirstResponder()
        }
    }

    func handleReactionButtonPress(sender: UIButton) {
        self.reactionLabel?.text = sender.currentTitle
        self.resignFirstResponder()
    }
}
```

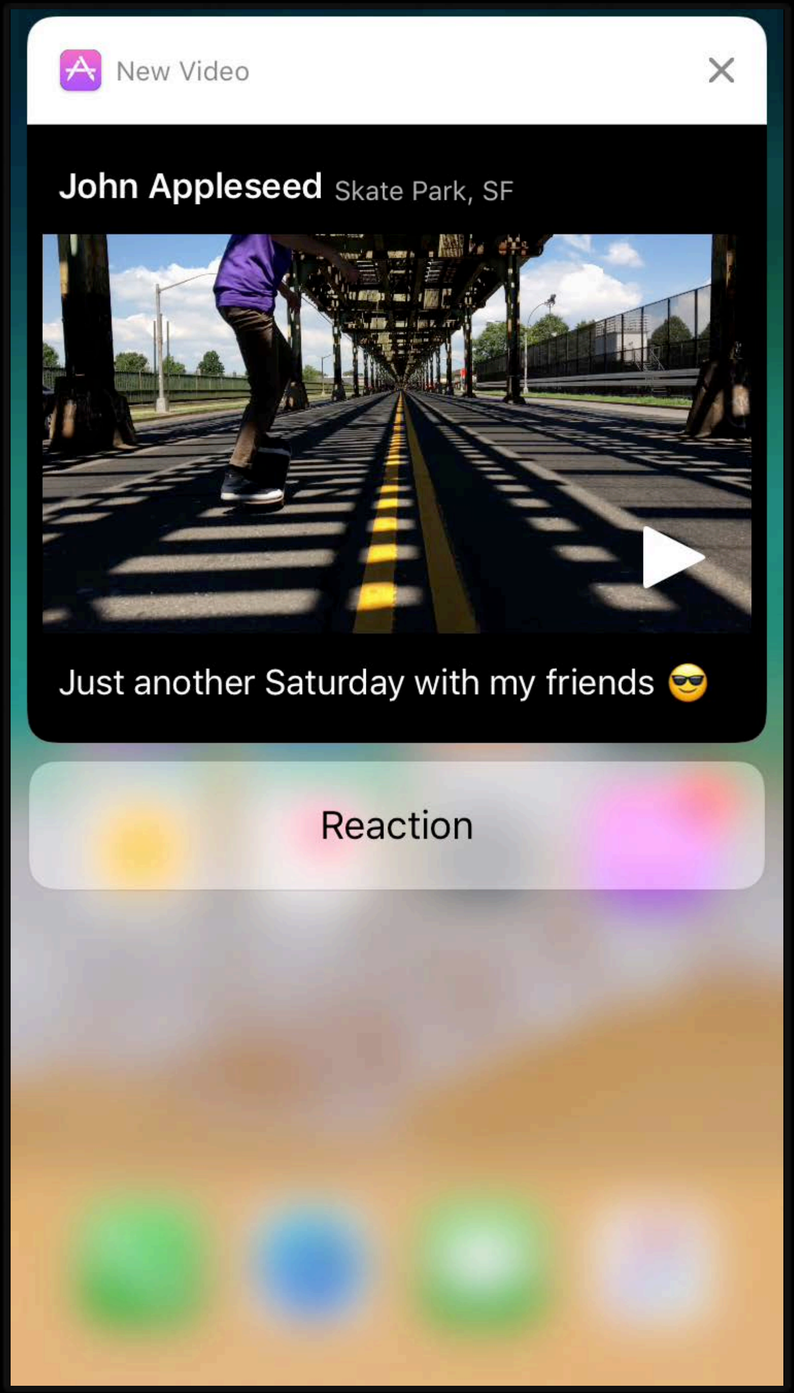
```
// User Input Customization — Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler
        completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "reactions-action" {
            self.becomeFirstResponder()
        }
    }

    func handleReactionButtonPress(sender: UIButton) {
        self.reactionLabel?.text = sender.currentTitle
        self.resignFirstResponder()
    }
}
```

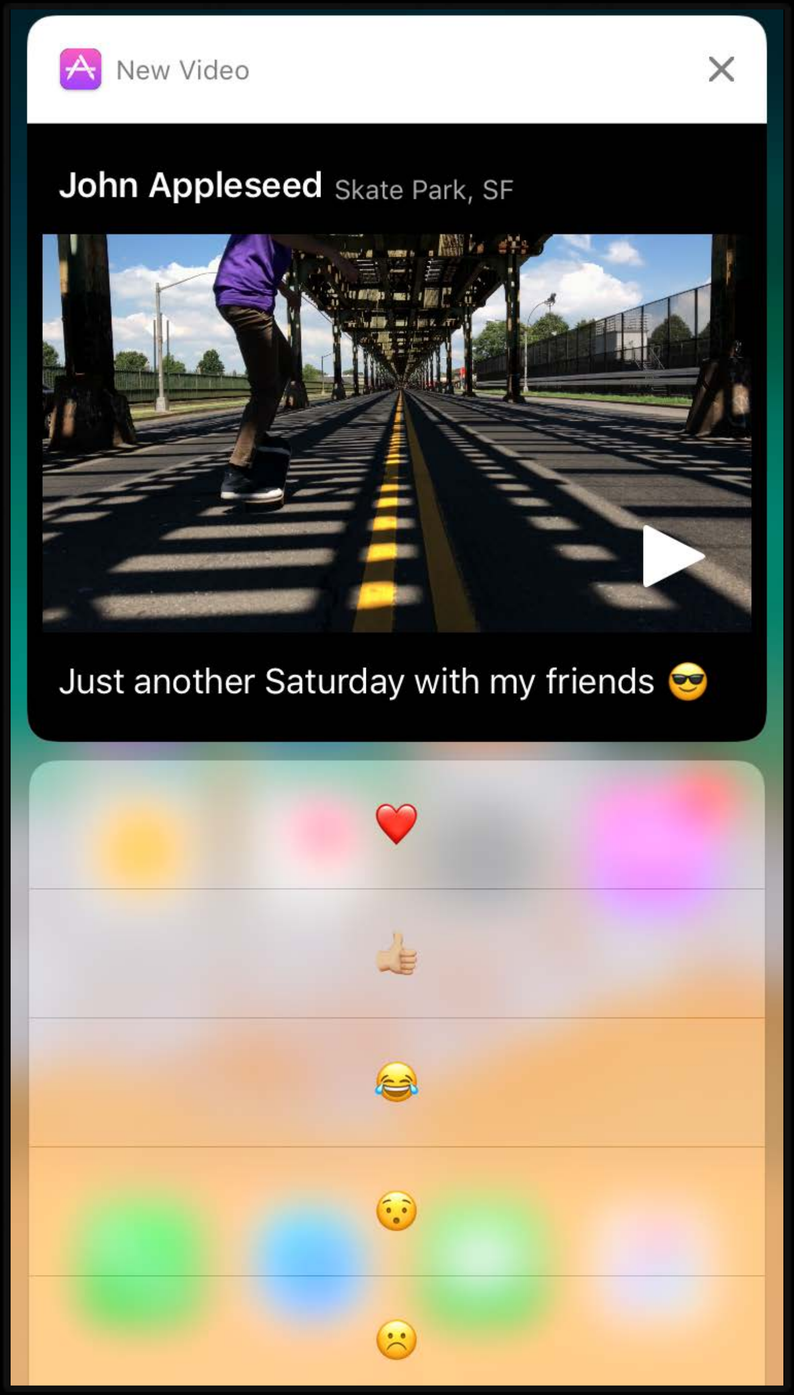
```
// User Input Customization — Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler
        completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "reactions-action" {
            self.becomeFirstResponder()
        }
    }

    func handleReactionButtonPress(sender: UIButton) {
        self.reactionLabel?.text = sender.currentTitle
        self.resignFirstResponder()
    }
}
```

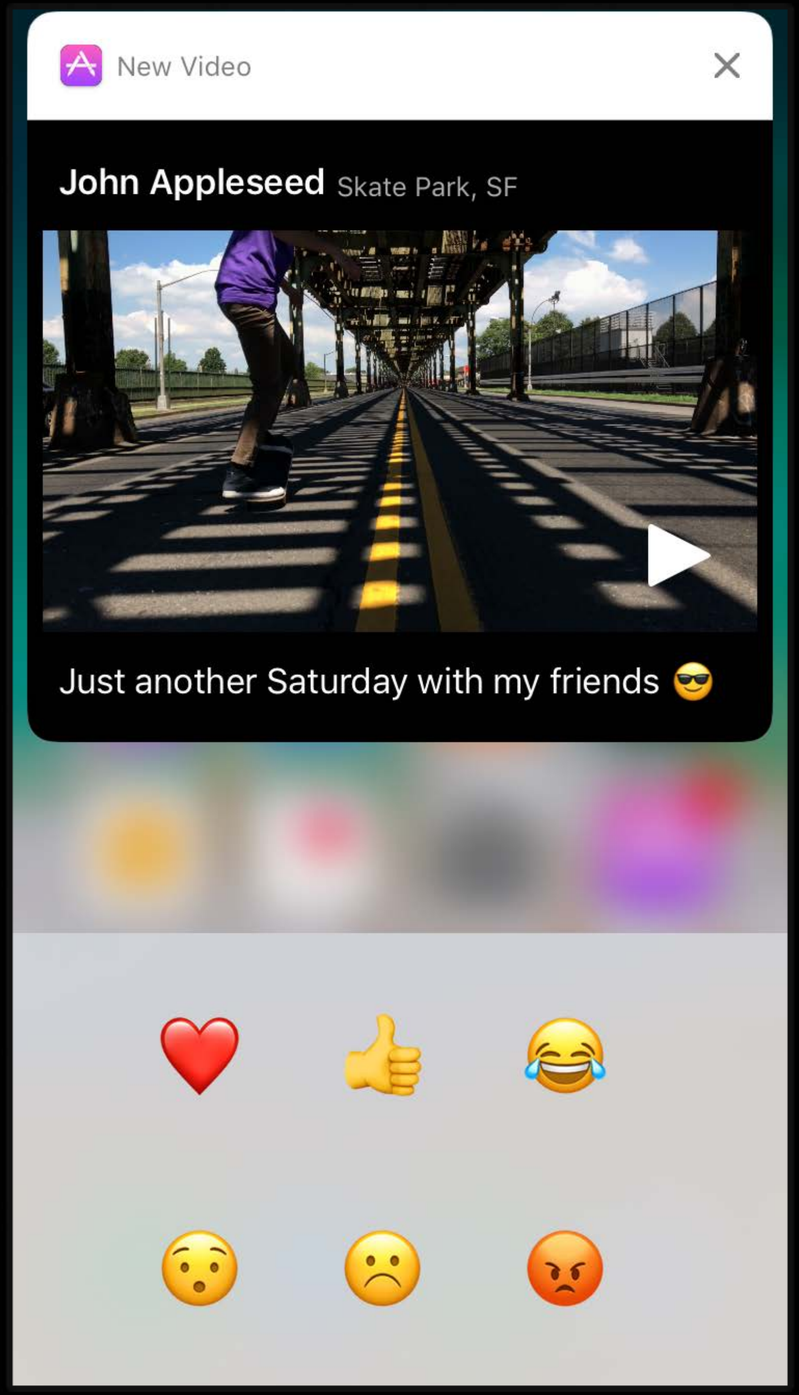
```
// User Input Customization — Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler
        completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "reactions-action" {
            self.becomeFirstResponder()
        }
    }

    func handleReactionButtonPress(sender: UIButton) {
        self.reactionLabel?.text = sender.currentTitle
        self.resignFirstResponder()
    }
}
```

```
// User Input Customization — Custom Input View

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler
        completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "reactions-action" {
            self.becomeFirstResponder()
        }
    }

    func handleReactionButtonPress(sender: UIButton) {
        self.reactionLabel?.text = sender.currentTitle
        self.resignFirstResponder()
    }
}
```


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
Reaction


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
Reaction


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎
❤️
👍🏼
😂
😯
☹️


New Video
John Appleseed Skate Park, SF
Just another Saturday with my friends 😎

User Input Customization

Best practices

User Input Customization

Best practices

User Input Customization

Best practices

Media buttons

User Input Customization

Best practices

Media buttons

Context specific actions

User Input Customization

Best practices

Media buttons

Context specific actions

To dismiss or not to dismiss

User Input Customization

Best practices

Media buttons

Context specific actions

To dismiss or not to dismiss

Custom user input view gives you full control

Showcase Apps

MIC

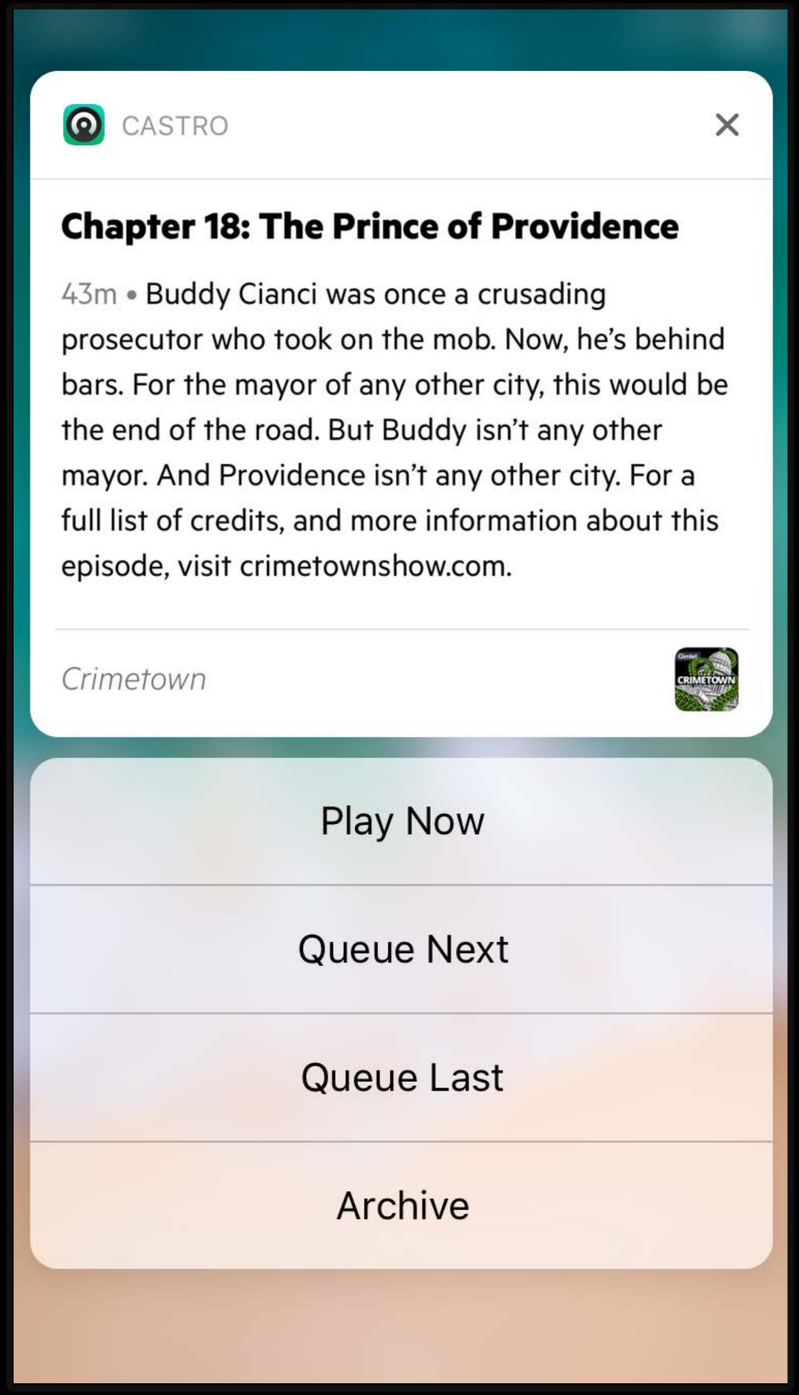
The 7 movies that took home the top prizes at the Cannes Film Festival

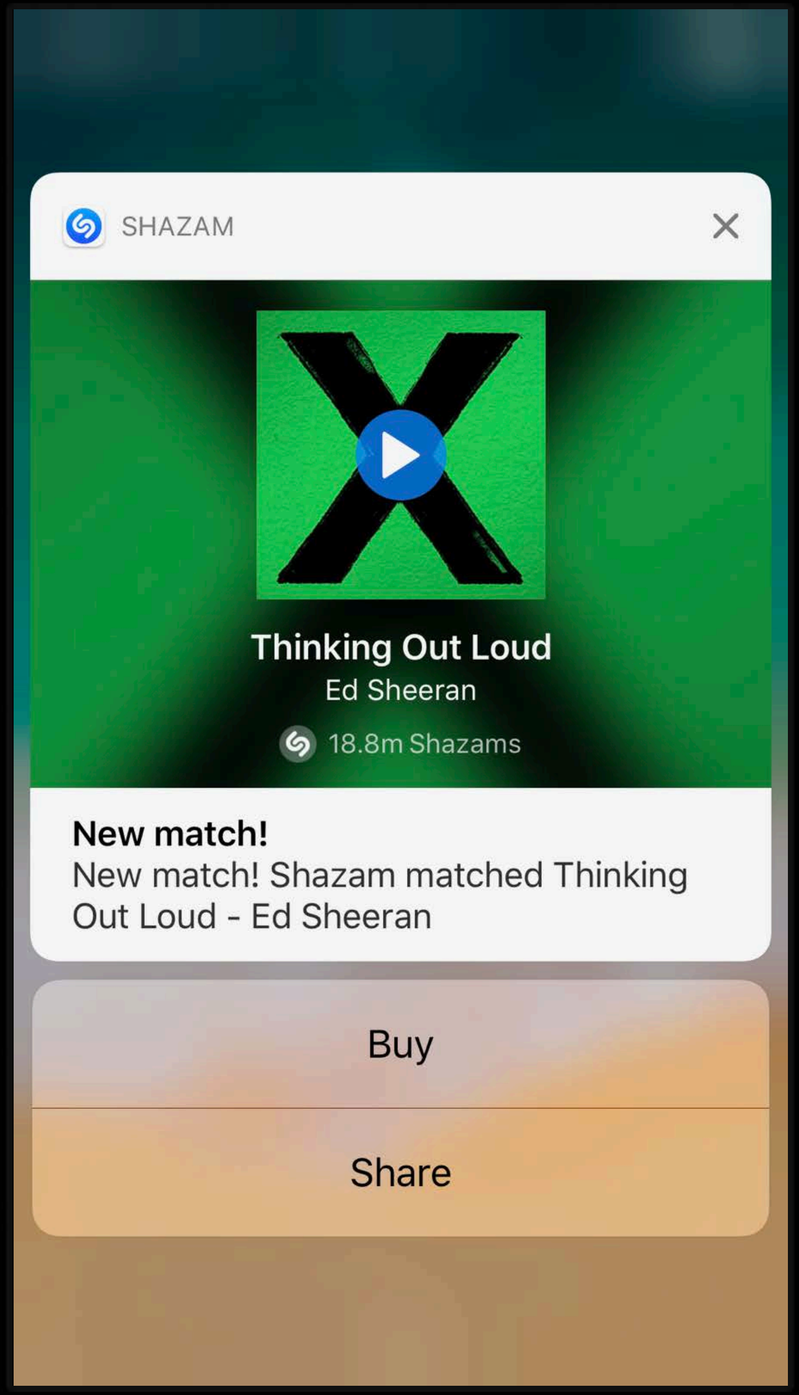
- 'The Square'
- '120 Beats Per Minute
- 'Loveless'
- 'The Beguiled'
- 'In the Fade'
- 'You Were Never Really Here'
- 'The Killing of a Sacred Deer'

MIC

The 7 movies that took home the top prizes at the Cannes Film Festival

- 'The Square'
- '120 Beats Per Minute
- 'Loveless'
- 'The Beguiled'
- 'In the Fade'
- 'You Were Never Really Here'
- 'The Killing of a Sacred Deer'


CASTRO
Chapter 18: The Prince of Providence
43m • Buddy Cianci was once a crusading prosecutor who took on the mob. Now, he's behind bars. For the mayor of any other city, this would be the end of the road. But Buddy isn't any other mayor. And Providence isn't any other city. For a full list of credits, and more information about this episode, visit crimetownshow.com.
Crimetown
CRIMETOWN
Play Now
Queue Next
Queue Last
Archive


SHAZAM
Thinking Out Loud
Ed Sheeran
18.8m Shazams
New match!
New match! Shazam matched Thinking Out Loud - Ed Sheeran
Buy
Share

Summary

Summary

Overview

Summary

Overview

Hidden notification content

Summary

Overview

Hidden notification content

Modifying content

Summary

Overview

Hidden notification content

Modifying content

Customizing rich notifications

Summary

Overview

Hidden notification content

Modifying content

Customizing rich notifications

User input customization

More Information

https://developer.apple.com/wwdc17/708

Labs

Notifications Lab	Technology Lab J	Thursday 1:50PM–4:00PM

WWDC17